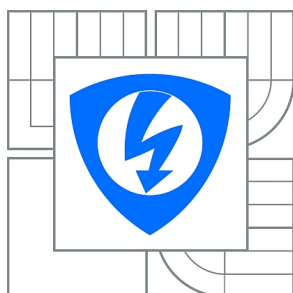


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNologiÍ**
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

WEBOVÁ APLIKACE PRO ODEVZDÁVÁNÍ STUDENTSKÝCH PRACÍ

WEB INTERFACE FOR SUBMITTING STUDENT PROJECTS

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. TOMÁŠ HILL

VEDOUcí PRÁCE
SUPERVISOR

Ing. TOMÁŠ PELKA

BRNO 2011



**VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ**

**Fakulta elektrotechniky
a komunikačních technologií**

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student: Bc. Tomáš Hill

ID: 102052

Ročník: 2

Akademický rok: 2010/2011

NÁZEV TÉMATU:

Webová aplikace pro odevzdávání studentských prací

POKYNY PRO VYPRACOVÁNÍ:

Úkolem studenta je v diplomové práci prostudovat možnosti webového frameworku web2py a navrhnout architekturu webové aplikace pro odevzdávání studentských projektů a pro instalaci virtuálních strojů v laboratoři PA126. Architektura bude dostatečně modulární a rozšiřitelná tak, aby výsledný produkt byl použitelný i v jiných laboratořích. Přesná specifikace systému bude součástí textu semestrálního projektu.

DOPORUČENÁ LITERATURA:

[1] DIPIERRO, Massimo. Web2py Enterprise Web Framework. 2 : Wiley, 2009. 344 s. ISBN 978-0470592359

[2] GOERZEN, John. Foundations of Python Network Programming. [s.l.] : Apress, 2004. 512 s. ISBN 978-1590593714

Termín zadání: 7.2.2011

Termín odevzdání: 26.5.2011

Vedoucí práce: Ing. Tomáš Pelka

prof. Ing. Kamil Vrba, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Diplomová práce se zabývá popisem možností webového frameworku Web2py. Webové frameworky mají za úkol usnadnit vývoj webových aplikací. Zvolený framework Web2py byl vytvořen v programovacím jazyce Python ve kterém probíhá i vývoj aplikací. Hlavní částí této diplomové práce je vytvoření webové aplikace pro správu projektů a evidenci docházky. Vytvořená aplikace je přizpůsobena potřebám laboratoře PA126.

KLÍČOVÁ SLOVA

Web2py, Framework, Webová aplikace, Python, DAL, XMLRPC, i18n

ABSTRACT

This master thesis deals with the description of web framework Web2py. The web frameworks are designed to facilitate the development of web applications. Selected web framework, Web2py, is written in Python programming language. The main purpose of this thesis is to create web application able to manage student projects and also manage students attendance. Created application is optimised for special needs of laboratory PA126.

KEYWORDS

Web2py, Framework, Web application, Python, DAL, XMLRPC, i18n

HILL, Tomáš *Webová aplikace pro odevzdávání studentských prací*: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2011. 53 s. Vedoucí práce byl Ing. Tomáš Pelka

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Webová aplikace pro odevzdávání studentských prací“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

Brno

.....

(podpis autora)

PODĚKOVÁNÍ

Děkuji vedoucímu diplomové práce panu Ing. Tomáši Pelkovi za podnětné rady a připomínky k vypracování této práce.

OBSAH

Úvod	9
1 Teoretický úvod	10
1.1 Převod objektů do relační databáze	10
1.1.1 Převod vztahů	11
1.1.2 Další techniky ORM	12
1.2 Python	12
1.2.1 Výkonnost	12
1.3 Web2py	13
1.3.1 Webový framework	14
1.3.2 Filozofie web2py	14
1.3.3 Bezpečnost	16
1.3.4 Jádro	18
1.3.5 Komunikace s databází	18
1.3.6 Metody pro řízení přístupu	20
2 Struktura aplikace	22
2.1 Návrh databáze	22
2.2 Adresářová struktura	23
2.3 Propojení s databází	24
2.4 Využití e-mailů	24
2.5 Překlad aplikace	26
2.6 Práce se soubory	26
2.7 Komunikace mezi aplikací a dalšími službami	27
3 Závěr	28
Literatura	30
Seznam symbolů, veličin a zkratk	31
Seznam příloh	33
A ER model databáze pro správu projektů	34
B Návod k používání webové aplikace	35
B.1 Modul Administrace	35
B.1.1 Uživatelé	36

B.1.2	Předměty	38
B.2	Modul Učitel	39
B.2.1	Projekty	40
B.2.2	Docházka	43
B.2.3	Souhrn hodnocení	44
B.3	Modul Student	44
B.3.1	Moje projekty	45
B.3.2	Hodnocení	47
C	Instalace frameworku Web2py a webové aplikace	49
C.1	Instalace na operačním systému Microsoft Windows	49
C.2	Instalace na operačním systému Linux	50
C.3	Instalace webové aplikace	51

SEZNAM OBRÁZKŮ

1.1	UML diagram objektu Student a jeho datový model	10
1.2	MVC – Model-View-Controller	14
1.3	Diagram sledu událostí podle návrhového vzoru MVC	15
1.4	Struktura frameworku Web2py	19
2.1	ER model databáze pro správu projektů	22
B.1	Úvodní obrazovka webové aplikace	35
B.2	Seznam uživatelů	36
B.3	Výsledek hromadně vytvářených uživatelů	37
B.4	Hromadné přiřazení studentů k vyučování	38
B.5	Hromadné přiřazení studentů k vyučování II	38
B.6	Hlavní strana po přihlášení učitele	39
B.7	Seznam projektů	40
B.8	Editace projektu	41
B.9	Komentáře k projektu	42
B.10	Virtuální stroje	43
B.11	Vytvoření nového virtuálního stroje	43
B.12	Evidence docházky	44
B.13	Souhrn hodnocení	45
B.14	Hlavní strana po přihlášení studenta	45
B.15	Seznam projektů studenta	46
B.16	Seznam virtuálních strojů přiřazených k projektu (modul Student) . .	47
B.17	Zobrazení souhrnu hodnocení studenta	48
C.1	Výchozí aplikace Welcome	52
C.2	Webové administrační rozhraní	53

ÚVOD

Tato diplomová práce se zabývá studiem webového frameworku web2py a jeho možnostmi při následném návrhu a provozu webové aplikace pro zadávání, odevzdávání a hodnocení studentských projektů a domácích úkolů.

Motivace

V současné době existuje pouze jediný systém pro správu předmětů a tím je e-learning (moodle ¹). Tento systém je bohužel naprosto nedostatečný, nepružný a nedostatečně modulární. Hlavní motivací pro vznik tohoto systému byla snaha vytvořit systém dostatečně obecný se speciálními funkcemi navíc, které naleznou využití především v předmětu Návrh, správa a bezpečnost počítačových sítí (dále MNSB). Jak bylo řečeno systém je dostatečně obecný i pro jiné předměty, ne jen MNSB. Hlavní předností oproti zmíněnému dosavadnímu řešení je skutečnost, že se soustředí na nedostatky současného řešení, tedy na proces odevzdání a bodování samostatných projektů.

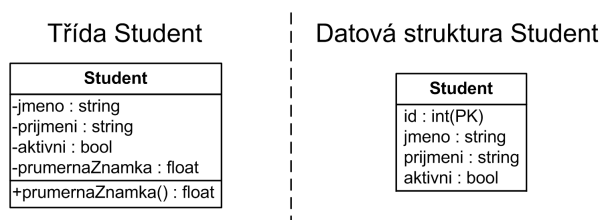
¹<http://moodle.org>

1 TEORETICKÝ ÚVOD

1.1 Převod objektů do relační databáze

V současné době probíhá vývoj většiny aplikací v objektově orientovaném prostředí (např. Java, C++, Python apod.) a zároveň existuje potřeba uchovávat data spojená s provozem těchto aplikací. V současnosti je nejlepším prostředkem pro uložení těchto dat relační databáze. Technika ORM (Object to Relation Mapping) je způsob jak správně převádět vlastnosti objektů a jejich vzájemné vztahy mezi těmito dvěma různými prostředími (objektové a relační).

V objektově orientovaném prostředí je základním prvkem objekt (instance třídy) a jeho atributy. Jednotlivé atributy jsou převedeny do relačního prostředí do žádného nebo více sloupců [1]. Jednoduchým příkladem může být objekt Student a jeho datový model v relační podobě (viz. UML diagram na obr. 1.1).



Obr. 1.1: UML diagram objektu Student a jeho datový model

Podle technik ORM je nyní potřeba vyrovnat rozdíly mezi oběma modely. Některé atributy, v tomto případě atribut `prumernaZnamka`, nebudou do datového modelu vůbec zahrnuty, neboť tyto atributy jsou vypočítávány aplikací a v databázi by byly zbytečné. Dále je nutné definovat tzv. „stínové“ atributy. Každý záznam (řádek) v tabulce relační databáze je jednoznačně identifikován pomocí čísla ID, který je ve většině případů také primárním klíčem tabulky. Pomocí primárních a cizích klíčů (tj. primárních klíčů jiných tabulek) jsou definovány vztahy mezi záznamy. Pokud načteme záznam z databáze do objektu, např. pro úpravu údajů, bylo by podle modelu na obr. 1.1 nemožné změnit také příslušný záznam v databázi - objekt v databázi pro úpravu není možné jednoznačně identifikovat. Proto je model objektu Student nutné rozšířit o atribut ID, který je stínovým atributem. Stínové atributy jsou použity pouze pro samotný převod mezi relačním a objektovým prostředím. Mezi další stínové atributy patří také různé časové otisky apod.

1.1.1 Převod vztahů

Další technikou ORM je vzájemný převod vztahů mezi objekty. Existují tři základní druhy vztahů:

- 1:1 - vztah mezi objekty je rovnocenný, první objekt odkazuje na druhý a naopak
- 1:N - objekt prvního typu je ve vztahu s více objekty druhého typu, naopak objekt druhého typu může být ve vztahu pouze s jedním objektem prvního typu
- M:N - více objektů prvního typu jsou ve vztahu s více objekty druhého typu a naopak

Vztahy mezi objekty můžeme také rozdělit do dalších dvou kategorií, na jednosměrné a obousměrné. U jednosměrného vztahu jeden z objektů neví o vztahu mezi nimi, u obousměrného ví o existenci vztahu oba objekty. Relační systém ale neumožňuje použití jednosměrných vztahů, proto musíme uvažovat veškeré vztahy jako obousměrné [1].

Implementace vztahů v objektech

Vztahy mezi objekty jsou reprezentovány v závislosti na typu vztahu. Při vzájemném vztahu 1:1 je použito metod „setter“, „getter“ a jednoho atributu u obou objektů. Vztah 1:N je vyjádřen opět metodami setter a getter, ale atribut je použitý pouze na straně objektu, jehož násobnost je 1. U objektu s násobností N je vztah vyjádřen kolekcí atributů, např. polem atributů Array. Analogicky u vztahu M:N je použita kolekce atributů o obou objektech.

Příkladem může být objekt Student s vazbou na objekt Trida typu 1:N. Objekt Student bude dle pravidel ORM obsahovat metody `getTrida()`, `setTrida()` a atribut `trida`. U objektu Trida budou definovány metody `getStudenti()`, `setStudenti()` a pole atributů `studenti`.

Implementace vztahů v relační databázi

Vyjádření vztahů v relační databázi je opět závislé na typu vztahu. Obecně se k vyjádření vztahů využívá cizích klíčů, což jsou primární klíče v jiných tabulkách. Vztah 1:1 je reprezentován cizím klíčem v jedné z tabulek, přitom nehraje roli ve které tabulce bude cizí klíč uložen - to je důsledek obousměrných vztahů v relační databázi. Vztah 1:N je možné vyjádřit dvěma způsoby: umístěním primárního klíče tabulky s násobností 1 jako cizí klíč do tabulky s násobností N, druhým způsobem je možné použití asociativní tabulky. Vyjádření vztahu M:N je možné provést také oběma způsoby, nicméně použití cizích klíčů přímo do tabulek by vedlo k velkému nárůstu počtu záznamů, proto je doporučeno používat spíše asociativní tabulku. Asociativní tabulka obsahuje cizí klíče z obou tabulek, M i N. Další sloupce nejsou nezbytně nutné, snižuje se tím tak množství duplicit, které by vznikly použitím cizích klíčů

přímo v tabulkách M a N. Asociativní tabulka se vřadí mezi tabulky M a N a vytvoří tak vztah 1:N na obě původní tabulky, kdy násobnost N je na straně asociativní tabulky [1].

1.1.2 Další techniky ORM

Do relační databáze lze také převést dedičnost mezi objekty, a to několika způsoby:

- Převod celého stromu objektů do jediné tabulky
 - použitelné pouze pro jednoduché stromy objektů.
- Každý objekt má vlastní tabulku
 - dedičnost je reprezentována vztahy mezi tabulkami, jedná se o mapování 1:1 z objektů do relační databáze.
- Převod do obecného schématu
 - používá se při vícenásobné dedičnosti a kombinuje obě metody převodu dedičnosti z objektů do relační databáze

ORM se také zabývá mapováním konstantních atributů tříd do relačního schématu nebo možnostmi ladění výsledného převodu z objektově orientovaného prostředí do relačního [1]. Tyto techniky jsou však již nad rámec tohoto textu.

1.2 Python

Python je interpretovaný programovací jazyk vyšší úrovně. Jazyk se postupem času stával stále populárnějším mezi programátory, především díky své jednoduché syntaxi [2]. Sám autor, Guido van Rossum, uvádí jako příklad programový kód ve vyšším programovacím jazyce o délce 20 000 řádků: „Kód o délce 20 000 řádků, vytvořený v jazyce Python, by byl pravděpodobně 100 000 řádků dlouhý, pokud by byl napsán v jazyce Java nebo C++ a dokonce až 200 000 řádků dlouhý pokud by se jednalo o jazyk C (díky chybějícím strukturám jazyků Java nebo C++)“ [2]. Úpravy kódu nebo hledání případných chyb je logicky snadnější v kódu o délce 20 000 řádků než v kódu o délce 100 000 řádků.

Python se řadí mezi dynamické funkcionální programovací jazyky se silným typováním. Zejména dynamika programovacího jazyka má vliv na výkonnost vykonávaného programu, viz. kap. 1.2.1.

1.2.1 Výkonnost

Jednoduchost syntaxe je vykoupena daní v podobě rychlosti vykonávání programu v jazyce Python. Jazyk lze zařadit do skupiny programovacích jazyků RAD (Rapid Application Development – Rychlý vývoj aplikace), jejichž překladače se starají

o mnoho aspektů programování, které musí být v tradičních programovacích jazycích explicitně ošetřeny [3]. Pokud chce programátor u tradičního programovacího jazyka, např. jazyka C, definovat proměnnou, např. pole, je nutné hned při deklaraci určit typ uložených dat v poli a velikost pole, nedochází tak ke zbytečnému plýtvání systémovými prostředky, nevýhodou však je že musíme dopředu znát rozsah pole. U jazyka Python stačí deklarovat pouze název proměnné, o vše ostatní se stará interpret až za běhu programu. Interpret poté musí sledovat, zda nedojde k překročení předem vymezené velikosti pole. Pokud ano, je nutné pole v paměti přesunout jinam kde je více místa, nebo tuto událost jinak ošetřit. Interpret také musí zjišťovat typ dat uložených v poli vzhledem k tomu, že se jedná o jazyk s dynamickou kontrolou typů. Z toho vyplývá jisté zpomalení běhu programu při výpočetně náročných operacích, u matematických operací je uváděn až desetinásobný a větší rozdíl mezi ekvivalentními programy v jazycích C a Python [3]. Je však nutno podotknout, že při dnešním výkonu výpočetní techniky se tato vlastnost projeví až u opravdu velkých projektů.

Tuto nevýhodu je možné odstranit spojením obou jazyků, pomocí modulů C/C++ propojených s jazykem Python. Výpočetně náročné aplikace tedy lze implementovat v jazyce C/C++, zbytek aplikace vyvíjet v jazyce Python a nakonec je propojit. Existují již i hotové moduly např. pro matematické operace, takže programátor nemusí přijít s jiným programovacím jazykem vůbec do styku.

Python je možné propojit s Javou, ovšem obráceně než jak je tomu u modulů pro jazyk C/C++. Kód jazyka Python je totiž implementován do kódu jazyka Java, následně kompilován do bajtového kódu, který je možné spustit v libovolné JVM (Java Virtual Machine – Virtuální stroj Java).

Obecně s ostatními programovacími jazyky lze Python propojit pomocí balíku COM (Component Object Model), např. s jazyky Visual Basic, Visual C++ nebo Delphi.

1.3 Web2py

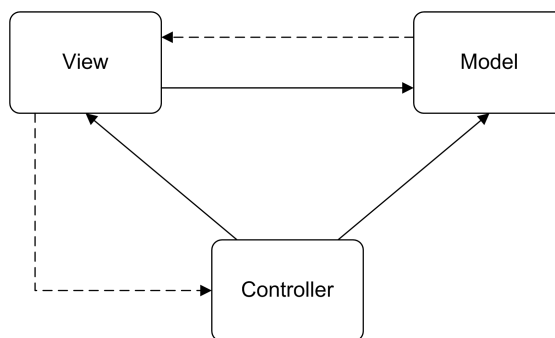
Web2py je webový framework napsaný v jazyce Python a v tomto jazyce jsou také programovány webové aplikace s pomocí Web2py. Projekt webového frameworku web2py byl zahájen vypuštěním první verze v roce 2007, autorem frameworku je Massimo Di Pierro [4]. V současné době na frameworku pracuje několik lidí současně, nedílnou součástí vývoje frameworku je také široká komunita programátorů, kteří mohou v diskusní skupině komunikovat přímo s autory frameworku a vyměňovat si tak své postřehy, upozorňovat na nalezené chyby ve frameworku a případně spolupracovat s autory na jejich odstranění.

1.3.1 Webový framework

Webový framework je v podstatě soubor nástrojů či funkcí, které shrnují nejčastěji používané postupy a metody, a slouží k pohodlnému a rychlému vývoji webové aplikace. Současně pomáhá programátorům vyvarovat se možných chyb, zejména v oblasti bezpečnosti a to právě zahrnutím obecně známých postupů řešení typických situací přímo do webového frameworku. Logickým důsledkem je také výrazné zkrácení výsledného programového kódu.

U webových frameworků se nejčastěji setkáme s architekturou MVC (Model-View-Controller). MVC je návrhový vzor, podle kterého se chová výsledná aplikace. Podstata MVC spočívá ve striktním oddělení logiky aplikace (Model), zobrazení (View), řízení a synchronizaci předcházejících dvou částí (Controller).

Na obr. 1.2 je znázorněn návrhový vzor MVC.



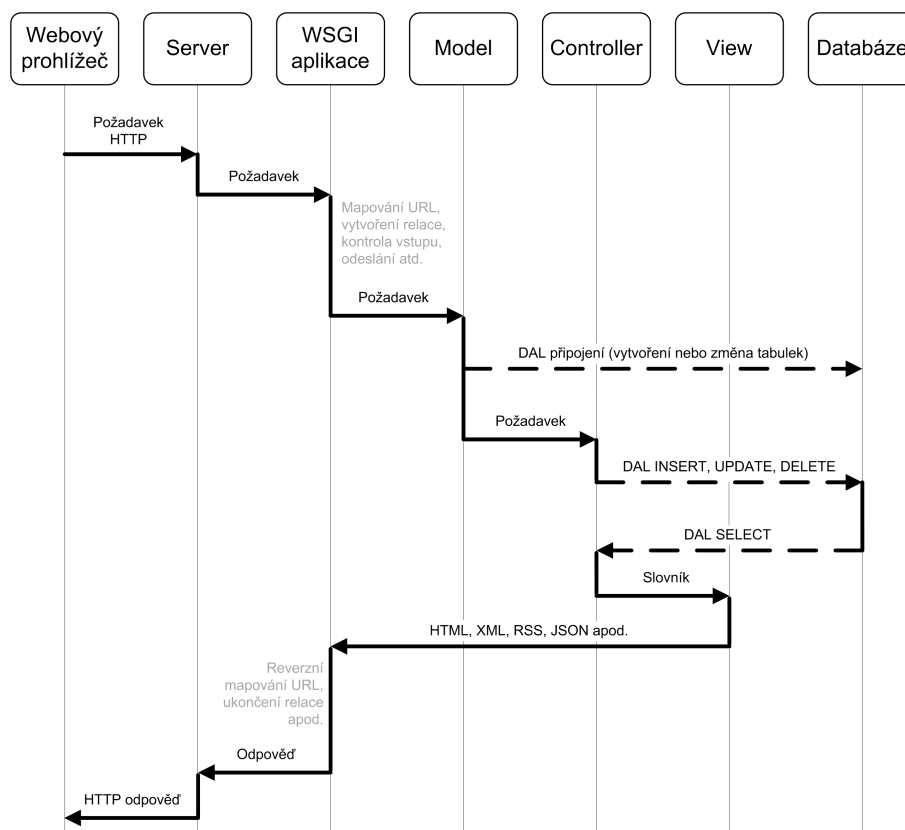
Obr. 1.2: MVC – Model-View-Controller

Typický průběh událostí ve webovém frameworku web2py podle návrhového vzoru MVC je znázorněn na obr. 1.3.

Webový prohlížeč odešle požadavek serveru, který zajišťuje přístup k aplikaci více uživatelům. Server předá požadavek danému vláknu aplikace, v diagramu WSGI aplikaci. Ta zajišťuje všechny běžné rutiny okolo běhu aplikace, spravuje cookies, relace, URL mapování, reverzní mapování apod. Trojice Model, View a Controller tvoří samotnou aplikaci. Posledním článkem v diagramu je databáze, kde jsou uložena všechna data. Připojení k databázi je řešeno pomocí DAL (Database Abstraction Layer – Abstraktní vrstva databáze) v části Model. Samotnou práci s daty již zajišťuje část Controller pomocí standardních SQL příkazů.

1.3.2 Filozofie web2py

Framework web2py přináší prostředí pro snadný a rychlý vývoj webových aplikací a v podstatě jde ve stopách jazyka Python: dosáhnout daného cíle s kódem,



Obr. 1.3: Diagram sledu událostí podle návrhového vzoru MVC

který je co možná nejpřehlednější a zároveň co nejkratší. V tomto směru jde web2py ještě dál, zapouzdřuje všeobecné postupy a metody do jediné funkce, čímž vlastně plní původní účel webového frameworku.

Jedním z cílů frameworku web2py je bezpečnost výsledné aplikace. Pomocí web2py bude programátor tvořit ve velké většině aplikace vyžadující přístup k databázi, s tím souvisí problematika autentizace a autorizace uživatele, ochrana proti útokům typu SQL injekce, útokům XSS (Cross-site Scripting) apod. Framework se snaží tyto procedury usnadnit a zároveň vyloučit chybu programátora při implementaci konkrétních postupů tak, aby byla aplikace co nejméně ohrožena z hlediska bezpečnosti. Více v kap. 1.3.3.

Nedílnou součástí frameworku je také DAL (Database Abstraction Layer – Abstraktní vrstva databáze). Tato vrstva přináší vyšší míru abstrakce pro práci s databázemi. V současné době DAL podporuje 9 různých databázových strojů [4], mezi nimiž je např. MySQL nebo Oracle, a maže tak možné rozdíly v používání SQL příkazů jednotlivými databázemi. Programátor jednoduše na začátku programování zavede ovladač pro příslušnou databázi a následně již používá jednotlivé funkce vrstvy DAL, které tak zastupují samotné SQL příkazy.

Autoři frameworku se také zavázali k zachování zpětné kompatibility v budoucích verzích. Znamená to, že aplikace napsaná v první vydané verzi web2py bude plně funkční i v té nejnovější verzi [4].

Web2py je vyvíjen jako svobodný software a distribuován pod licencí GPL 2.0. Nicméně výsledné aplikace nejsou nijak vázány touto licencí a mohou tak být mj. zkompileovány do bajtkódu a distribuovány jako uzavřený, komerční produkt. Výjimku tvoří případ, kdy by aplikace obsahovala byť jen kus kódu frameworku web2py, poté by se na aplikaci vztahovala původní licence GPL tak jako na samotný framework.

1.3.3 Bezpečnost

Při vývoji aplikace se programátor může dopustit neúmyslných chyb, které mohou vést k ohrožení bezpečnosti aplikace. Hrozí tak zneužití chyb programátora např. k získání dat z databáze či jiné využití aplikace ve svůj prospěch. Cílem frameworku web2py je zabránit těmto chybám. Pro ilustraci následuje seznam deseti nejrizikovějších oblastí či problémů při vývoji bezpečné aplikace, který sestavila organizace zabývající se bezpečností aplikací OWASP (Open Web Application Security Project). U každého bodu je uveden způsob, jakým se framework s tímto problémem vypořádá [4].

- Cross-site Scripting (XSS): problém se týká neošetřených vstupních dat z webového formuláře. Útočník může aplikaci podstrčit svůj kód, který může způsobit např. „krádež“ relace uživatele, změnu vzhledu a chování webové aplikace apod.
 - Web2py ve výchozím nastavení kontroluje všechny vstupy proti tomuto typu útoku.
- Injekce škodlivého kódu: toto se týká především SQL injekce, kdy je útočníkem vložen do regulárního dotazu SQL další SQL příkaz, který může v důsledku obejít veškerá zabezpečení a získat tak např. data z databáze nebo je upravovat či mazat.
 - Databáze ve frameworku web2py je obsluhována vrstvou DAL, která znemožňuje použití SQL injekce. SQL příkazy totiž nejsou definovány programátorem přímo v kódu aplikace, ale jsou dynamicky generovány přímo vrstvou DAL, která injekci ošetřuje.
- Přímé odkazy na objekty: k problému dochází, pokud programátor odkáže přímo na vnitřní implementaci objektu (např. soubor, složka apod.) v podobě odkazu URL nebo parametru formuláře.
 - Framework nikdy neodhaluje vnitřní strukturu (objekty), navíc při každém požadavku je zkontrolována adresa URL jestli je v požadovaném

tvaru. Tím se také předchází možnému útoku na strukturu adresářů.

- Cross Site Request Forgery (CSRF): útok je založen na podstrčení škodlivé stránky přihlášenému uživateli, která provede útočníkem definovanou akci.
 - Framework tento typ útoku řeší generováním jednorázového tokenu pro každou akci a znemožňuje tak tento typ útoku.
- Nevhodné zpracování chyb: další možnou chybou vývojáře je vyzrazení interního chování aplikace např. při chybě, kdy se vypíše aplikační kód přímo uživateli do prohlížeče.
 - Web2py obsahuje systém zaznamenávání chyb pomocí tzv. lístkového systému („ticketing system“). Při výskytu chyby se daná chyba zapíše do souboru a uživateli je vypsán do prohlížeče pouze kód události. Zdrojový kód tak zůstane přístupný pouze administrátorům.
- Špatná správa relací: nastává v případě nevhodné správy tokenů relací, kdy útočník může získat hesla, klíče nebo autentizační tokeny a vydávat se tak za někoho jiného.
 - Framework obsahuje vestavěný mechanismus přihlášení administrátorů, kdy spravuje relace nezávisle mezi aplikacemi. Administrační rozhraní také vynucuje použití zabezpečených cookies. Samotné aplikace jsou chráněny pomocí metod založených na RBAC (Role-Based Access Control), ve frameworku web2py tyto metody implementuje třída `Auth`, viz. kap. 1.3.6.
- Nezabezpečené uchování dat: problém se týká především hesel uložených v databázi. Pokud útočník nějakým způsobem získá přístup k databázi a zároveň není implementována žádná z metod šifrování hesel, útočník má okamžitě přístup k heslům v textové podobě.
 - Web2py v základním nastavení podporuje hash algoritmy MD5 a HMAC+SHA-512 pro uchování hesel v databázi. Současně také umožňuje použití dalších algoritmů pro šifrování hesel.
- Nezabezpečená komunikace: mnoho aplikací nepoužívá zabezpečenou komunikaci mezi serverem a klientem ani v nejnútnejších případech, např. při přenosu hesla od klienta k serveru (autentizace).
 - Web2py obsahuje webový server Rocket WSGI s podporou SSL (Secure Socket Layer). Framework také umožňuje použití webových serverů Apache nebo Lighttpd s modulem `mod_ssl` (modul zajišťuje podporu SSL).
- Ovládání aplikace pomocí URL: mnoho aplikací je ovládáno pouze pomocí předem definovaných URL. Zabezpečení je řešeno pouze ve smyslu skrytí těchto URL v prohlížeči uživatele. Útočník může této chyby využít a získat přístup k aplikaci bez autentizace.
 - Web2py provádí mapování URL požadavků k příslušným modulům a funk-

cím. Dále také jasně definuje, které funkce jsou veřejné a které jsou přístupné pouze po autentizaci a autorizaci. Tyto opatření zamezí ovládání aplikace pouhou znalostí URL.

1.3.4 Jádru

Framework se skládá ze čtyř základních celků [4], viz. obr. 1.4:

- Aplikace
- Knihovny jádra
- Contrib
- Webový server Rocket WSGI

Část „Aplikace“ obsahuje jednotlivé aplikace. V balíku Web2py jsou standardně obsaženy aplikace admin, examples a welcome. Aplikace admin slouží ke správě ostatních aplikací pomocí webového rozhraní. Pomocí aplikace admin a webového prohlížeče je také možné provádět testy a zpřístupnit příkazovou řádku přímo ve webovém prohlížeči pro ladění aplikace. Aplikace examples obsahuje dokumentaci a ukázkové aplikace. Aplikace welcome je kostrou pro aplikace vytvořené ve web2py, obsahuje základní menu a přihlašovací stránku pro autentizaci uživatele. Součástí celku „Aplikace“ jsou také případné zásuvné moduly - pluginy.

Knihovny jádra jsou základem frameworku Web2py a sdružují všechny funkcionality frameworku.

Část „Contrib“ obsahuje další knihovny, které byly vyvinuty třetími stranami a byly zahrnuty do základního balíku Web2py.

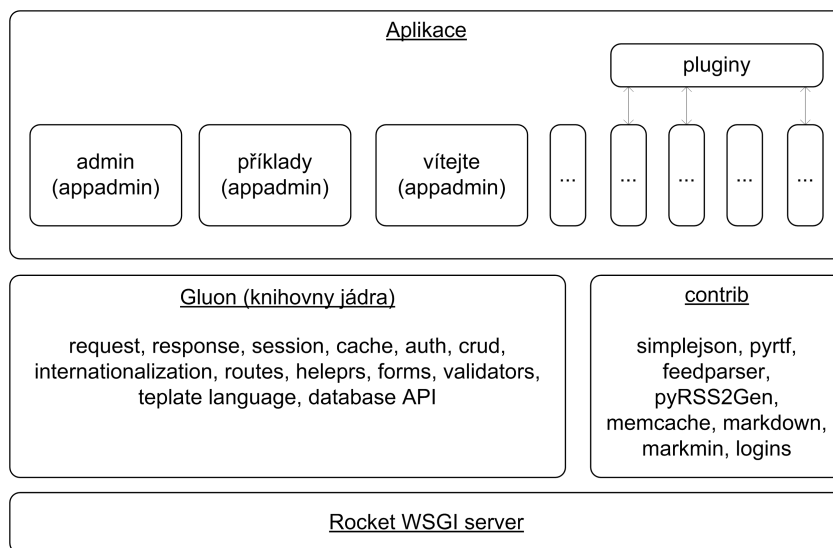
Poslední částí frameworku je webový server Rocket WSGI s nativní podporou SSL.

1.3.5 Komunikace s databází

Ve frameworku Web2py je komunikace s databází řešena abstraktní vrstvou DAL (Database Abstraction Layer – Abstraktní vrstva databáze). DAL v současné době podporuje následující databáze:

- SQLite
- PostgreSQL
- MySQL
- Oracle
- * a další, viz. [4]

Ve skutečnosti vrstva DAL nekomunikuje přímo s databázemi, ale příkazy předává prostřednictvím externích ovladačů. V distribuci frameworku Web2py je standardně nainstalován ovladač pro databázi SQLite (tato databáze je také součástí distribuce).



Obr. 1.4: Struktura frameworku Web2py

Pokud programátor chce pracovat s jinou databází než SQLite, je nutné doinstalovat příslušný ovladač.

Pro použití v kódu aplikace jsou definovány funkce abstraktní vrstvy DAL, které provádějí příslušné akce nad databází. Základní konstrukcí je připojení k databázi:

```
db = DAL('sqlite://storage.db')
```

Tato konstrukce zajistí připojení k databázi, ke které je možné od této chvíle přistupovat přes proměnnou `db`. Po připojení k databázi je možné pomocí abstraktní vrstvy DAL provádět dotazy, vytvářet tabulky nebo celé databáze, vkládat data atd. Následující kód představuje příklad vytvoření entity s názvem „Osoby“ se sloupci „Jmeno“ a „Prijmeni“ (řádek 1) a následně vložení záznamu (řádek 2).

```
db.define_table('Osoby', Field('Jmeno'),
                Field('Prijmeni'))
db.person.insert(Jmeno="Josef", Prijmeni="Novák")
```

Ve skutečnosti se vytvoří entita s třemi sloupci: „Jmeno“, „Prijmeni“ a „id“. Sloupec „id“ se vytvoří zcela automaticky jako primární klíč, číslo, které je automaticky zvyšováno s každým novým záznamem. To umožňuje jednotlivé záznamy od sebe jednoznačně odlišit. „id“ se také používá pro vazby mezi entitami.

Rozsáhlá knihovna funkcí abstraktní vrstvy DAL obsahuje např. vnitřní i vnější JOIN příkazy, příkazy pro tvorbu virtuálních entit, vytváření vazeb apod. Pokud by programátorovi tyto příkazy nestačily, je možné s pomocí abstraktní vrstvy DAL použít standardní SQL příkaz, např.:

```
db.executesql('SELECT * FROM Osoby;')
```

Pomocí příkazu `executesql` lze také využívat funkce specifické pro konkrétní databázový stoj. Příkladem může být konstrukce `ILIKE` v databázi PostgreSQL, která na rozdíl od standardního SQL příkazu `LIKE` nerozlišuje malá a velká písmena (není tzv. case sensitive). Pokud je to ale možné, je lepší se používání příkazu `executesql` vyhnout. Vstupy použité v této funkci nejsou kontrolovány na možný škodlivý kód, tzn. databáze není ochráněna proti SQL injekci (což je jedna z výhod vrstvy DAL). Další problém při použití této funkce spočívá v případné změně databázového stroje. Pokud bychom totiž využívali specifických funkcí databázového stroje, je velice pravděpodobné, že jiný databázový stroj tyto funkce nebude podporovat a všechny tyto funkce by se musely upravit pro nový databázový stroj.

Přístup frameworku Web2py k databázi pomocí abstraktní vrstvy DAL a ovladače databáze přináší dvě zásadní výhody:

- Jednoduchá rozšiřitelnost
 - Pro podporu nového typu databáze stačí vyvinout ovladač databáze, který bude kompatibilní s abstraktní vrstvou DAL.
- Snadná změna databáze
 - Pokud je programátor nucen z nějakého důvodu změnit databázový stroj, stačí pouze nainstalovat ovladač nového databázového stroje a změnit příkaz vykonávající připojení k databázi - `DAL('sqlite://storage.db')`. V kódu aplikace není nutné měnit žádný kód díky vrstvě DAL a jejích abstraktních funkcích, které provádějí operace s daty v databázi. Jak již ale bylo řečeno, problém může nastat při používání specifických funkcí v příkazu `executesql`.

1.3.6 Metody pro řízení přístupu

Pro řízení přístupu k funkcím aplikace, tedy autentizaci a autorizaci, implementuje framework Web2py třídu `Auth`, která zajišťuje řízení přístupu na základě metody RBAC (Role-Based Access Control).

RBAC je model řízení přístupu k aplikaci a jejím funkcím za pomoci relační databáze. V databázi jsou uloženi jednotliví uživatelé, jejich hesla a každému uživateli mohou být přiděleny různé úrovně oprávnění. Při přihlášení je tedy nejdříve ověřena identita uživatele pomocí uživatelského jména a hesla (autentizace). Následně je pro každou akci vykonanou uživatelem ověřeno, zda má uživatel přidělené požadovanou úroveň oprávnění pro danou akci (autorizace).

Pro funkčnost řízení přístupu vyžaduje třída `Auth` v databázi tyto tabulky [4]:

- `auth_user`
 - Tabulka obsahuje jednotlivé uživatele, jejich e-mailové adresy, hesla a stav, ve kterém se nachází (aktivní, blokový nebo v přechodu).

- `auth_group`
 - V této tabulce jsou uživatelé přiřazeni do skupin, tabulka je propojena vztahem M:N s tabulkou uživatelů `auth_user`. Každý uživatel je standardně členem své vlastní skupiny, ale může být členem i dalších skupin. Podobně skupina může obsahovat více uživatelů.
- `auth_membership`
 - Tato tabulka je asociativní tabulkou pro vztah M:N mezi tabulkami `auth_user` a `auth_group`.
- `auth_permission`
 - Tabulka přiděluje skupinám (`auth_group`) jednotlivá oprávnění. Standardně stačí definovat pouze název oprávnění, nicméně je možné záznamy rozšířit o položky `table` a `record`, které dále upřesňují jaký druh operace pro konkrétní tabulku je skupině přidělen.
- `auth_event`
 - V této tabulce jsou evidovány veškeré změny v ostatních tabulkách a uskutečněné přístupy k objektům z databáze (CRUD).

Po definici těchto tabulek je programátorovi dostupné jednoduché ale účinné rozhraní pro ověřování identity uživatele a jeho přístupových práv. Tyto tabulky lze dokonce přejmenovat podle potřeb programátora a to přepsáním příslušných proměnných hodnot na začátku kódu (po definici objektu `Auth`).

Jednotné přihlášení

Autentizace uživatele za pomoci tabulky v relační databázi není jediným možným způsobem ověření identity. Framework Web2py dovoluje využít pro registraci a autentizaci služeb třetích subjektů, např. OpenID, Facebook, LinkedIn, Google, MySpace, Flickr, LDAP apod. [4].

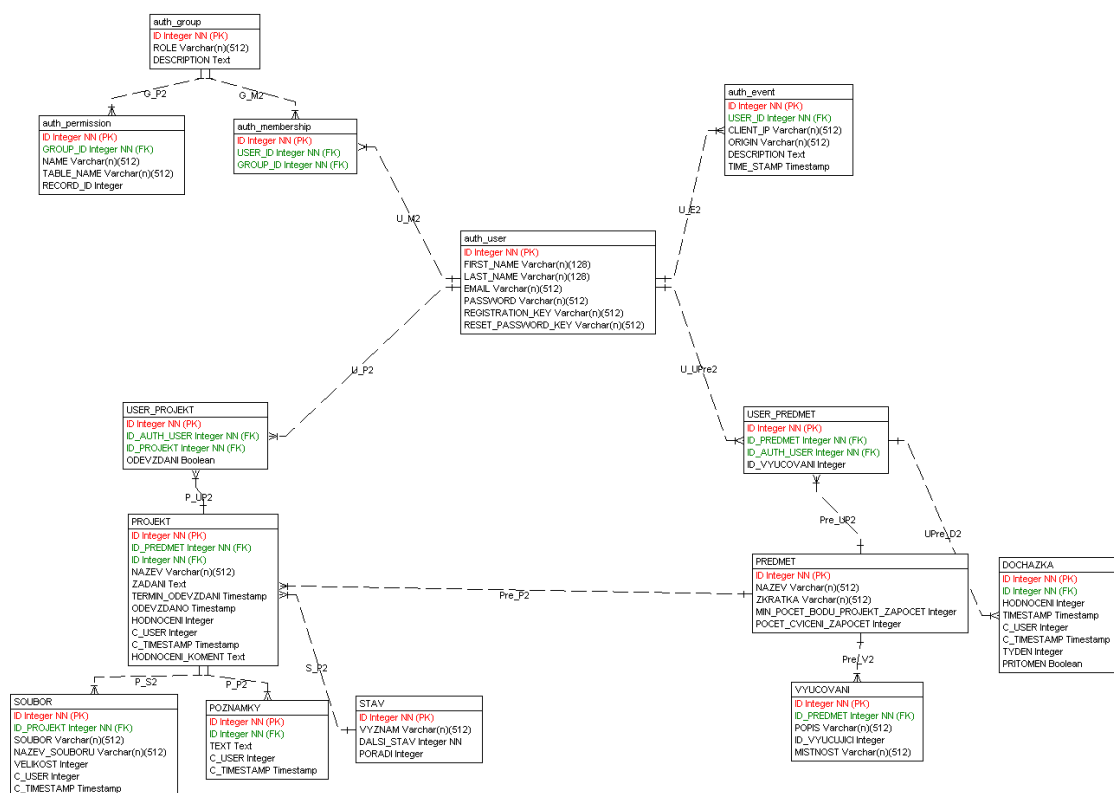
Nejjednodušším řešením je použití pluginu od společnosti Janrain. Implementace je skutečně snadná, nejdříve je nutné se zaregistrovat na webových stránkách společnosti Janrain, která vygeneruje jedinečný klíč pro implementaci do webové aplikace [5]. Programátor poté zajistí načtení pluginu do aplikace a zadá vygenerovaný klíč. Toto řešení má dvě zásadní nevýhody: jsou podporovány jenom světově známé služby, např. Facebook, Google, Twitter apod. [5]. Např. není podporován protokol LDAP, tzn. není možné přes službu Janrain provést přihlášení k soukromému LDAP serveru. Druhou a podstatnou nevýhodou je to, že údaje pro přihlášení jsou předávána službám přes prostředníka - server Janrain.

Další možností přihlášení je přímé připojení ke službě bez prostředníka. Framework Web2py v současnosti standardně podporuje autentizaci např. pomocí SMTP protokolu (např. Gmail nebo jiný SMTP server), služby OpenID nebo přihlášení k LDAP serveru [4] pomocí pluginů jednotlivých služeb. Nevýhodou oproti řešení s pluginem Janrain je nastavení pluginu pro každou službu zvlášť.

2 STRUKTURA APLIKACE

2.1 Návrh databáze

Hlavním úkolem webové aplikace je správa projektů a virtuálních strojů, které jsou k projektům přiřazeny, aplikace ale také slouží k evidenci docházky studentů do vyučování. Pro uchování dat souvisejících s provozem aplikace slouží relační databáze, jejíž ER model je znázorněn na obr. 2.1 nebo v příloze A.



Obr. 2.1: ER model databáze pro správu projektů

První částí modelu databáze je pět entit sloužících ke správě uživatelů, jejich přiřazení do skupin a přidělení příslušných oprávnění daným skupinám. Jsou to tyto entity:

- auth_user
- auth_group
- auth_membership
- auth_permission
- auth_event

Názvy entit byly zachovány v souladu s frameworkem Web2py, nicméně tyto entity lze dodatečně přejmenovat. Význam jednotlivých entit byl popsán v kapitole 1.3.6.

Entita `auth_user` je propojena vztahy M:N s entitami `PROJEKT` a `PREDMET` (přes vazební tabulky `USER_PROJEKT` a `USER_PREDMET`).

V entitě `PROJEKT` jsou uchovány domácí úkoly nebo projekty jednotlivých uživatelů a jsou rozlišeny pomocí cizího klíče `ID_PREDMET` do kterého předmětu daný projekt patří. K projektu mohou být přiřazeny soubory (entita `SOUBOR`) a komentáře (entita `POZNAMKY`). Parametry a informace o virtuálních strojích (specifický požadavek předmětu `MNSB`) jsou udržovány v externí databázi, tyto informace jsou získávány pomocí `XMLRPC` (viz. kap. 2.7). Jmenované entity (`SOUBOR` a `POZNAMKY`) jsou propojeny s entitou `PROJEKT` vztahem typu 1:N. Entita `STAV` vyjadřuje slovní popis stavu, ve kterém se projekt nebo úkol nachází (vytvořený, zadaný, odevzdaný nebo ohodnocený). V jednotlivých částech webové aplikace jsou vytvořena pravidla, kdo a kdy může vykonávat určité operace s projektem a ve kterém stavu se musí projekt nacházet, aby operace byla povolena.

Entita `PREDMET` uchovává názvy předmětů spolu s jejich zkratkami. Na tuto entitu je napojena vztahem 1:N entita `VYUCOVANI`, kde jsou definovány vyučovací hodiny pro jednotlivé předměty. Entita slouží především pro snadnější vyhledávání (např. výpis seznamu studentů předmětu `MNSB` v hodině Středa 11h-13h). Entita `PREDMET` je za pomoci vazební entity `USER_PREDMET` propojena s entitou `auth_user`. Ve vazební entitě se nachází atribut `ID_VYUCOVANI`. Tento atribut umožňuje zařazení studenta v rámci předmětu do konkrétní vyučovací hodiny. Na tuto vazební entitu je dále napojena entita `DOCHAZKA`, která slouží k evidenci docházky studentů do vyučování a případné hodnocení studenta v rámci vyučovací hodiny.

2.2 Adresářová struktura

Aplikace se skládá z několika souborů jejichž struktura respektuje model MVC (viz. kap. 1.3.1) [4]. Soubory lze rozdělit do pěti základních kategorií:

- **models:** popisují reprezentaci dat
- **controllers:** popisují logiku aplikace a tok dat
- **views:** popisují reprezentaci dat
- **languages:** soubory překladu do jiných jazyků
- **modules:** moduly jazyka Python používané v aplikaci
- **static:** obrázky, CSS soubory stylů apod.

2.3 Propojení s databází

Aplikace komunikuje s databází pomocí abstraktní vrstvy DAL (viz. kap. 1.3.5). Framework Web2py v sobě obsahuje základní databázový stroj SQLite, požadavkem pro vypracování diplomové práce byla komunikace s databázovým strojem PostgreSQL. Změna databázového stroje spočívá ve změně ovladače databázového stroje (viz. „Instalace frameworku Web2py a webové aplikace“, příloha C).

Ve zdrojovém kódu aplikace je připojení definováno v souboru `models/db.py`:

```
db = DAL('postgres://web2py:123@localhost/projekty')
```

Význam jednotlivých částí parametru funkce DAL:

- `postgres`: definice typu databázového stroje
- `web2py`: přihlašovací jméno pro přístup do databáze
- `123`: heslo pro přístup do databáze
- `localhost`: umístění databázového stroje v síti
- `projekty`: název databáze

Jednotlivé tabulky a jejich vzájemné relační vztahy udržuje framework na základě zdrojového kódu v souboru `models/db.py`. Po deklaraci databázového připojení lze vytvářet tabulky, definovat jejich atributy a relační vztahy mezi tabulkami přímo v tomto souboru, při instalaci či přesunu aplikace na nový databázový stroj musí být v databázovém stroji definována pouze nová databáze, strukturu a relační vztahy si vytvoří framework sám při prvním přístupu k aplikaci. Například vytvoření tabulek PROJEKT, PREDMET a relačního vztahu mezi nimi typu 1:N je definováno následovně:

```
#deklarace tabulky PROJEKT a jejích atributů
db.define_table('PROJEKT',
    Field('ID_PREDMET', db.PREDMET),
    Field('ID_STAV', db.STAV),
    Field('NAZEV'),
    Field('ZADANI', 'text'),
    Field('TERMIN_ODEVZDANI', 'datetime'),
    Field('ODEVZDANO', 'datetime'),
    Field('HODNOCENI', 'integer', default=0),
    Field('HODNOCENI_KOMENT', 'text'),
    Field('C_USER', 'integer'),
    Field('C_TIMESTAMP', 'datetime'))

#deklarace tabulky PREDMET a jejích atributů
db.define_table('PREDMET',
    Field('NAZEV'),
    Field('ZKRATKA'),
    Field('POCET_CVICENI_ZAPOCET', 'integer', default=13),
    Field('MIN_POCET_BODU_PROJEKT_ZAPOCET', 'integer', default=0))

#deklarace relačního vztahu 1:N
db.PROJEKT.ID_PREDMET.requires = IS_IN_DB(db, db.PREDMET.id, '%(ZKRATKA)s')
```

2.4 Využití e-mailů

Webová aplikace vytvořená pomocí frameworku Web2py má mnoho možností spojení s autentizací uživatele, např. lze povolit či zakázat registraci uživatelů do sys-

tému, povolit registraci uživatelů za pomoci registračního e-mailu (po zaregistrování musí uživatel registraci potvrdit pomocí odkazu, který mu je zaslán na e-mailovou adresu uvedenou při registraci) apod. Aplikace pro správu projektů a evidenci docházky má v tomto směru tyto vlastnosti:

- registrace uživatelů je zakázána (uživatele vytváří administrátor),
- změna dat v profilu uživatelů je zakázána (změny provádí pouze administrátor),
- uživatel může v případě ztráty hesla požádat o nové heslo (odesílá se e-mail na uživatelskou e-mailovou adresu)

Zdrojový kód zajišťující požadované vlastnosti (`models/db.py`):

```
auth.settings.actions_disabled.append('register')
auth.settings.actions_disabled.append('profile')
auth.settings.reset_password_requires_verification = True
```

Pro podporu vlastností spojených s odesíláním e-mailů musí být nejprve definováno připojení k smtp serveru (`models/db.py`):

```
from gluon.tools import Mail
mail = Mail(globals())
mail.settings.server = 'smtp.server.com:25'
mail.settings.sender = 'emailova_adresa@server.com'
mail.settings.login = 'emailova_adresa@server.com:heslo'
auth.settings.mailer = mail
```

Odesílání e-mailů je využíváno kromě žádosti o nové heslo (v případě zapomenutí) také v následujících případech:

- zadání nového projektu (e-mail se odesílá všem řešitelům),
- vytvoření nového komentáře k projektu (e-mail se odesílá všem řešitelům projektu a také zadavateli),
- ohodnocení projektu (e-mail se odesílá všem řešitelům).

Pro odesílání e-mailů se používají šablony uložené ve složce `sablony`, při odesílání jsou nahrazeny proměnné uvozené v šabloně znakem `$` skutečnými hodnotami. Zdrojový kód pro odeslání e-mailové zprávy v případě zadání nového komentáře:

```
slovník = {'predmet':predmet, 'nazev_projektu':nazev_projektu, 'deadline':deadline, 'komentar':form.vars.TEXT,\
          'autor':autor.first_name + ' ' + autor.last_name}
for i in resitele:
    vysledek = odeslat_email(i.auth_user.email, slovník, 'komentar_zadan.tpl')
vysledek = odeslat_email(zadavatel_email, slovník, 'komentar_zadan.tpl')

def odeslat_email(email, parametry, nazev_sablony):
    import string, os
    # parametry: dict pro nahrazení údajů v šabloně
    try:
        cesta = os.path.join(request.folder, 'sablony', nazev_sablony)
        sablona = open(cesta, 'r')
    except:
        return T('System is not able to open the template.')

    mailBody = ''
    radek = ''
    predmet = ''
    telo = False
    while 1:
        if not telo:
            predmet = sablona.readline()
            telo = True
```

```

radek = sablona.readline()
if not radek:
    break
mailBody += radek
sablona.close()
mailBody = string.Template(mailBody).safe_substitute(parametry)
predmet = string.Template(predmet).safe_substitute(parametry)
mail.send(to=email, subject=predmet, message=mailBody)
return 0

```

2.5 Překlad aplikace

Jádro frameworku Web2py obsahuje vestavěný překladač (i18n - Internationalization), jehož pomocí lze snadno vytvořit aplikaci s podporou více jazyků. Při vytváření zdrojového kódu se využívá vestavěného objektu `T`, kterým je veškerý text překládán podle aktuálně nastaveného jazyka [4]. Překlad se provádí pouhým zaměňováním slovních spojení umístěných jako parametr objektu `T` za spojení v daném jazykovém souboru. Z toho vyplývá, že po vytvoření aplikace je nutné provést manuální překlad těchto slovních spojení.

Doporučením autora frameworku je psát slovní spojení standardně v anglickém jazyce (viz. [4]), neboť pokud není nalezen požadovaný jazykový soubor, překlad se nezdaří a je zobrazeno slovní spojení tak jak bylo definováno v objektu `T`. Jednotlivé jazykové soubory se nacházejí ve složce `languages` pod svou zkratkou (např. `languages/cs-cs.py`). Samotná volba jazyka probíhá na základě informací HTTP hlavičky nebo vynuceně pomocí příkazu umístěném např. v `models/db.py`:

```
T.force('cs-cs')
```

Webová aplikace vytvořená v rámci této diplomové práce dodržuje doporučení autora frameworku Web2py, tzn. veškerá slovní spojení jsou psána ve zdrojovém kódu v anglickém jazyce a následně byl vytvořen český překlad v souboru `languages/cs-cs.py`. Pro překlad lze použít administrační rozhraní frameworku, kde jsou přehledně označeny nová, ještě nepřeložená spojení.

2.6 Práce se soubory

Jednou z funkcí webové aplikace je také možnost nahrávat soubory na server a přiřazovat je k jednotlivým projektům. Soubory jsou nahrány do složky `uploads`, přejmenovány z důvodu ochrany proti útoku typu procházení adresáře (directory traversal attack) a nové jméno souboru je uloženo do databáze. Všechny tyto úkony jsou zajištěny samotným frameworkem, programátor definuje pouze tabulku v databázi, kam se budou informace o souborech ukládat. Příklad definice tabulky pro nahrávání souborů na server v souboru `models/db.py`:

```
db.define_table('SOUBOR',
    Field('ID_PROJEKT', db.PROJEKT),
    Field('NAZEV_SOUBORU'),
    Field('SOUBOR', 'upload'))
```

Atribut `SOUBOR.SOUBOR` musí být definován jako datový typ `upload`, ve skutečnosti se do databáze do atributu `SOUBOR` uloží nový název souboru po přejmenování, tak jak je uložen na serveru ve složce `uploads`, např. `SOUBOR.SOUBOR.a9a4d394d883889b.706964617931302d68702e6769666.gif`. Původní název souboru je uložen do atributu `NAZEV_SOUBORU` pro následnou práci se souborem (např. zobrazení názvu v seznamu souborů, stažení souboru atd.).

2.7 Komunikace mezi aplikací a dalšími službami

Webová aplikace umožňuje vytváření a automatickou nebo manuální instalaci virtuálních strojů, dále také jejich spouštění a zastavování. Všechny tyto funkce jsou řešeny pomocí vzdáleného volání procedur (XMLRPC), kdy webová aplikace figuruje jako klient připojený k serveru a využívá vzdáleně jeho funkce. Ve zdrojovém kódu aplikace je nutné načíst potřebnou knihovnu a definovat připojení k serveru (socket, uživatelské jméno a heslo). Poté již lze volat funkce na serveru přes definované připojení:

```
import xmlrpclib
PROTOKOL='https'
JMENO='manager'
HESLO='heslo'
ADRESA='147.229.151.221'
PORT=65500
url = '%s://%s:%s@%s:%s' % (PROTOKOL, JMENO, HESLO, ADRESA, PORT)

pripojeni = xmlrpclib.ServerProxy(url)
vysledek = pripojeni.funkce_na_serveru()
```

3 ZÁVĚR

Diplomová práce popisuje webový framework Web2py pro tvorbu webových aplikací. Byly rozebrány důležité oblasti při tvorbě webových aplikací a jejich řešení ve frameworku Web2py. Ve druhé části diplomové práce byly popsány stěžejní části webové aplikace pro správu projektů a evidenci docházky z pohledu zdrojového kódu. Samotná webová aplikace je dostupná na přiloženém disku CD v komprimované podobě připravené přímo pro instalaci na webový server (`instalace/web2py.app.projekty.w2p`) a také jako samostatné zdrojové soubory (`adresář aplikace`). Struktura disku CD:

- **aplikace**: zdrojové soubory webové aplikace.
- **dokumentace**: dokumentace řídicích funkcí ve formě HTML souborů.
- **instalace**: instalační soubory pro instalaci webového frameworku Web2py, databázového stroje PostgreSQL a ovladače databáze pro operační systémy Microsoft Windows a Linux.

V příloze A je uveden entitně-relační model (ER model) databáze, kde jsou uložena provozní data aplikace. ER model databáze byl detailně popsán v kap. 2.1. Další příloha B popisuje používání aplikace z pohledu uživatele (správce, učitel a student). Poslední příloha diplomové práce C obsahuje postup instalace webového frameworku Web2py, databázového stroje PostgreSQL a samotné webové aplikace na operační systémy Microsoft Windows a Linux.

Vlastnosti vytvořené webové aplikace:

- Správa projektů:
 - Zadávání, odevzdání a hodnocení projektů.
 - Jeden projekt může být přiřazen více studentům (práce v týmu).
 - Možnost nahrát na server soubory s vypracovaným projektem.
 - Podpora komentářů k projektu.
 - Projekty jsou rozlišeny podle předmětu do kterého spadají, studentem který projekt vypracovává a zadávajícím učitelem.
 - Vytváření, instalace, spouštění či zastavování virtuálních strojů - není součástí této diplomové práce, v aplikaci je řešena pouze komunikace se serverem spravujícím virtuální stroje pomocí XMLRPC a prezentace výsledků.
- Evidence docházky studentů do vyučování a jejich hodnocení
- Omezení přístupových práv:
 - Vyučující může hodnotit pouze projekty, které sám zadal (stejný princip platí i u docházky).
 - Studentům jsou zobrazeny pouze projekty které jsou jim přiděleny.
 - Učitelům jsou zobrazeny pouze projekty které sami zadali.

Další vlastnosti implementované pomocí frameworku Web2py:

- Odesílání e-mailové notifikace v následujících případech:
 - Registrace uživatele do systému.
 - Zadáání projektu k vyřešení.
 - Vytvoření nového komentáře k projektu.
 - Hodnocení projektu.
- Hromadné vytváření uživatelů na základě souboru typu csv (přizpůsobeno pro export uživatelů ze systému Apollo VUT).

Způsob tvorby webové aplikace ve frameworku Web2py umožňuje aplikaci kdykoliv rozšířit o další moduly zajišťující specifické požadavky jiných předmětů, čemuž by ovšem podléhala také odpovídající změna ER modelu databáze. Provedená realizace aplikace splňuje specifické požadavky pro výuku v laboratoři PA126 a lze ji také okamžitě využít i v ostatních předmětech a laboratořích pro zadávání projektů a evidenci docházky studentů.

LITERATURA

- [1] AMBLER, Scott W. *Agile Database Techniques : Effective Strategies for the Agile Software Developer*. Canada : Wiley, 2003. 480 s. ISBN 0-471-20283-5.
- [2] VENNERS, Bill. *Artima* [online]. 27.1.2003 [cit. 2010-10-26]. Programming at Python Speed. Dostupné z WWW: <<http://www.artima.com/intv/speed.html>>.
- [3] HARMS, Daryl; MCDONALD, Kenneth. *Začínáme programovat v jazyce Python*. 2. opravené vydání. Brno : Computer Press, 2008. 472 s. ISBN 978-80-251-2161-0.
- [4] DIPIERRO, Massimo. *The Official web2py Book* [online]. 3rd Ed. c2010 [cit. 2010-11-11]. Programming at Python Speed. Dostupné z WWW: <<http://web2py.com/book/>>.
- [5] *Janrain* [online]. c2010 [cit. 2010-12-01]. Supported Networks. Dostupné z WWW: <<http://www.janrain.com/solutions/supported-networks>>.
- [6] *Linux Packages* [online]. c2011 [cit. 2011-05-19]. PostgreSQL. Dostupné z WWW: <<http://www.postgresql.org/download/linux>>.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

CNRI Corporation for National Research Initiatives

COM Component Object Model

CRUD Create, Read, Update, Delete

CSRF Cross Site Request Forgery

CWI Centrum Wiskunde & Informatica

DAL Database Abstraction Layer – Abstraktní vrstva databáze

GPL General Public License

HTML HyperText Markup Language

HTTP Hyper Text Transfer Protocol

i18n Internationalization

JSON JavaScript Object Notation

JVM Java Virtual Machine – Virtuální stroj Java

LDAP Lightweight Directory Access Protocol

MVC Model-View-Controller

ORM Object to Relation Mapping

OWASP Open Web Application Security Project

PSF Python Software Foundation

RAD Rapid Application Development – Rychlý vývoj aplikace

RBAC Role-Based Access Control

RHEL Red Hat Enterprise Linux

RPC Remote Procedure Calls

RSS Really Simple Syndication

SQL Structured Query Language

SSL Secure Socket Layer

UML Unified Modeling Language

URL Uniform Resource Locator

WSGI Web Server Gateway Interface

XML Extensible Markup Language

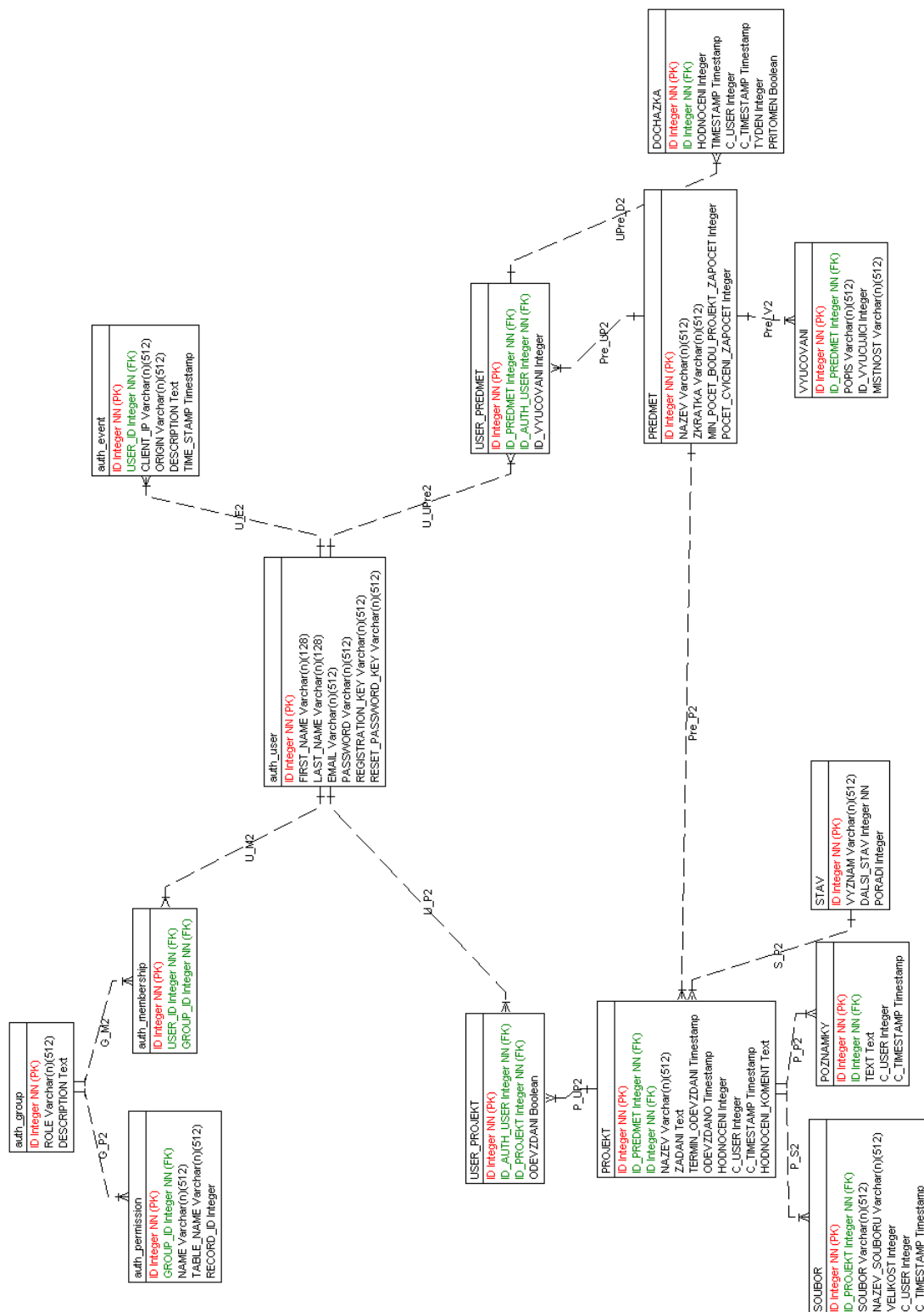
XMLRPC Extensible Markup Language - Remote Procedure Calls

XSS Cross-site Scripting

SEZNAM PŘÍLOH

A	ER model databáze pro správu projektů	34
B	Návod k používání webové aplikace	35
B.1	Modul Administrace	35
B.1.1	Uživatelé	36
B.1.2	Předměty	38
B.2	Modul Učitel	39
B.2.1	Projekty	40
B.2.2	Docházka	43
B.2.3	Souhrn hodnocení	44
B.3	Modul Student	44
B.3.1	Moje projekty	45
B.3.2	Hodnocení	47
C	Instalace frameworku Web2py a webové aplikace	49
C.1	Instalace na operačním systému Microsoft Windows	49
C.2	Instalace na operačním systému Linux	50
C.3	Instalace webové aplikace	51

A ER MODEL DATABÁZE PRO SPRÁVU PROJEKTŮ



B NÁVOD K POUŽÍVÁNÍ WEBOVÉ APLIKACE

Spuštěním aplikace ve webovém prohlížeči dojde k načtení hlavní stránky (viz. obr. B.1). Aplikace nedovoluje pracovat bez přihlášení, proto je nutné se nejdříve přihlásit. Přihlášením dojde k načtení práv daného uživatele a k načtení příslušných modulů. V systému mohou být uživateli přiřazeny role **student**, **učitel** nebo **správce**, v závislosti na přidělených rolích jsou načteny moduly **Student**, **Učitel** nebo **Administrace**. Bez přihlášení lze pouze zvolit jazyk používaný v aplikaci.

K přihlášení do aplikace je nutné zadat přihlašovací jméno (e-mailová adresa) a heslo. V případě že uživatel své heslo zapomene, lze po ověření identity (pomocí e-mailové adresy) vytvořit heslo nové.



Obr. B.1: Úvodní obrazovka webové aplikace

B.1 Modul Administrace

Struktura modulu Administrace:

- Uživatelé
 - Seznam uživatelů
 - Nový uživatel
 - Noví studenti (ze souboru csv)
- Předměty
 - Seznam předmětů
 - Nový předmět
 - Seznam vyučování
 - Nové vyučování

B.1.1 Uživatelé

Sekce **Uživatelé** umožňuje zobrazení seznamu všech uživatelů, dále umožňuje vytvořit nové uživatele jednotlivě nebo hromadně pomocí souboru csv. Po zvolení položky **Seznam uživatelů** dojde k zobrazení filtru, který dokáže omezit seznam uživatelů podle zadaných parametrů. Odesláním formuláře dojde k zobrazení seznamu uživatelů (viz. obr. B.2), pokud uživatel nezadá žádnou položku filtru, dojde k vypsání všech uživatelů. Kliknutím na ID uživatele se načte formulář pro změnu dat uživatele, případně lze v tomto formuláři uživatele odstranit. V tomto formuláři jsou také přidělovány či odebírány role uživatele.

Pokud v aplikaci dojde k výpisu seznamu (např. seznam uživatelů, seznam projektů apod.), lze tyto položky řadit pomocí parametrů, které jsou v záhlaví tabulky zvýrazněny jako odkazy a to pomocí kliknutí na dané záhlaví (např. Příjmení). Pod seznamem se nachází navigační lišta - aktuální strana seznamu, pohyb mezi stranami seznamu, zobrazené položky, počet položek celkem a také je zde možnost zvolit počet záznamů na jedné straně.

ID	Jméno	Příjmení	E-mail	Počet předmětů
432	Pavel	Dvořák	xxdvora07@stud.feec.vutbr.cz	0
433	Jiří	Friedberg	xxfried04@stud.feec.vutbr.cz	0
434	Petr	Grobelný	xxgrobe00@stud.feec.vutbr.cz	0
435	Václav	Hauser	xxhause01@stud.feec.vutbr.cz	0
436	Michal	Hlavica	xxhlavi00@stud.feec.vutbr.cz	0
437	Michal	Holiš	xxholis00@stud.feec.vutbr.cz	0
438	Petr	Hošek	xxhosek07@stud.feec.vutbr.cz	0
439	Oldřich	Hrabčík	xxhrabc05@stud.feec.vutbr.cz	0
440	Michal	Hrubý	xxhruby00@stud.feec.vutbr.cz	0
441	Martin	Hruška	xxhrusk00@stud.feec.vutbr.cz	0

Obr. B.2: Seznam uživatelů

Odkaz **Nový uživatel** načte formulář pro vytvoření nového uživatele. Po odeslání formuláře je uživatel přesměrován automaticky na editaci vytvořeného uživatele, kde lze případně provádět změny nebo přidávat či odebírat role v systému. Nově vytvořenému uživateli se automaticky přiřazuje role **student**.

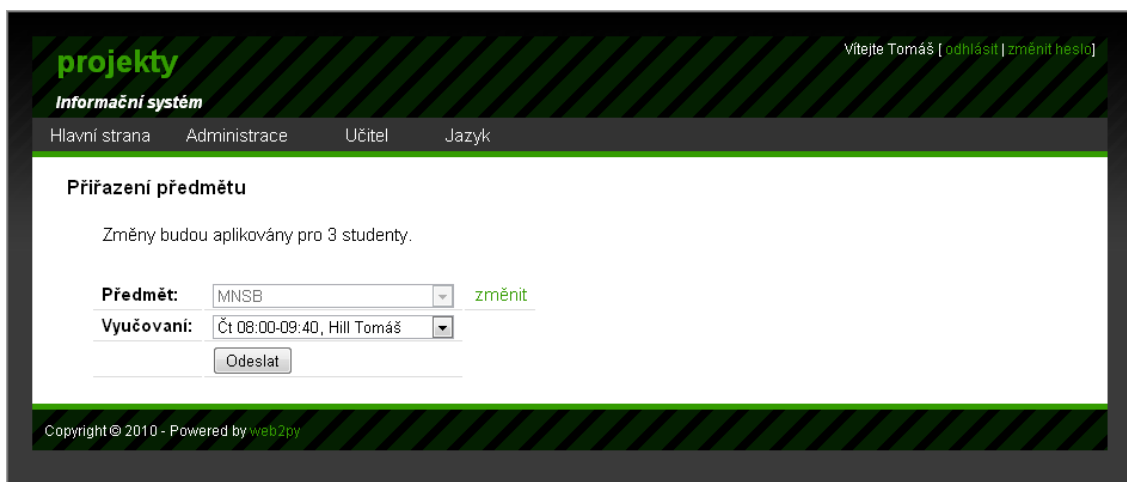
Další částí modulu **Administrace** a sekce **Uživatelé** je možnost hromadného vytváření studentů pomocí souboru csv. Po kliknutí na odkaz **Noví studenti** (ze souboru csv) se načte jednoduchý formulář, kde uživatel zvolí soubor csv k nahrání na server. Položky v souboru csv musí být ve formátu "login";"příjmení";"jméno", přičemž na prvním řádku je hlavička (tzn. první řádek je systémem ignorován). Ve formuláři se kromě volby souboru nachází také položka **Doména**, kde je předvyplněno **stud.feec.vutbr.cz**. Položkou login ze souboru csv a zde vyplněnou doménou se vytvoří e-mailová adresa uživatele a uloží se do databáze. Po odeslání formuláře je soubor na serveru vyhodnocen a pokud vyhovuje požadovanému formátu, dojde k vytvoření uživatelů (viz. obr. B.3). Pokud uživatel v systému již existuje (resp. v systému existuje e-mailová adresa), uživatel není vytvořen a toto je oznámeno ve výpisu výsledku hromadného vytváření uživatelů.

Příjmení	Jméno	E-mail	Výsledek
Havel	Jiří	xxhavel00@stud.feec.vutbr.cz	Vytvořený
Doležal	Martin	xxdolezal00@stud.feec.vutbr.cz	Selhalo (uživatel již existuje)
Doubek	Lukáš	xxdoubek00@stud.feec.vutbr.cz	Vytvořený
Černocký	Petr	xxcerno00@stud.feec.vutbr.cz	Selhalo (uživatel již existuje)
Hejduk	Pavel	xxhejduk01@stud.feec.vutbr.cz	Vytvořený

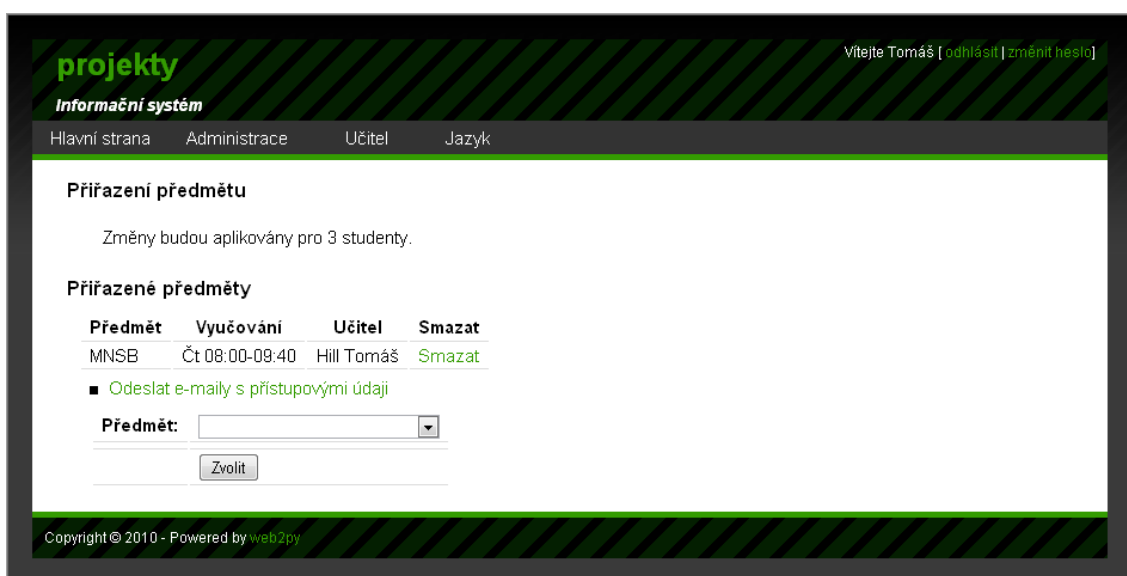
Obr. B.3: Výsledek hromadně vytvářených uživatelů

Po hromadném vytvoření uživatelů lze tyto nově vytvořené uživatele přiřadit k vyučováním (viz. obr. B.4) pomocí odkazu **Možnosti pro tyto uživatele**.

Po přiřazení minimálně jednoho vyučování je možné odeslat přístupové údaje do systému pomocí odkazu **Odeslat e-mail s příkazovými údaji** (viz. obr. B.5). V rámci hromadného zakládání uživatelů dochází i k vygenerování náhodného hesla pro každého uživatele. Toto heslo se skládá z malých písmen, velkých písmen a číslic, délka hesla odpovídá počtu 10 znaků. Na závěr jsou heslo, přístupové jméno a registrované předměty zaslány pomocí šablony jednotlivým uživatelům na jejich e-mailové adresy.



Obr. B.4: Hromadné přiřazení studentů k vyučování



Obr. B.5: Hromadné přiřazení studentů k vyučování II

B.1.2 Předměty

V další sekci, modulu **Předměty**, lze zobrazit **Seznam předmětů** nebo je možné založit **Nový předmět**. Funkčně se tyto dvě funkce shodují se seznamem uživatelů, resp. vytvořením nového uživatele, včetně možnosti filtrace podle názvu předmětu nebo podle zkratky předmětu. Při zakládání nového předmětu se vyplňují kromě názvu předmětu a jeho zkratky také **počet cvičení pro obdržení zápočtu** a **minimální počet bodů ze všech projektů**. Pomocí těchto dvou atributů je ověřováno, zda má student nárok na zápočet.

Funkce **Nové vyučování** zakládá vyučování předmětu, pokud tedy chce administrátor vytvořit vyučování, musí mít definovaný minimálně jeden předmět, kterému vyučování přiřadí. Ve formuláři lze vybrat učitele, resp. uživatele s přidělenou rolí **učitel**, který bude mít přístup k tvorbě docházky, projektům a jejich hodnocení.

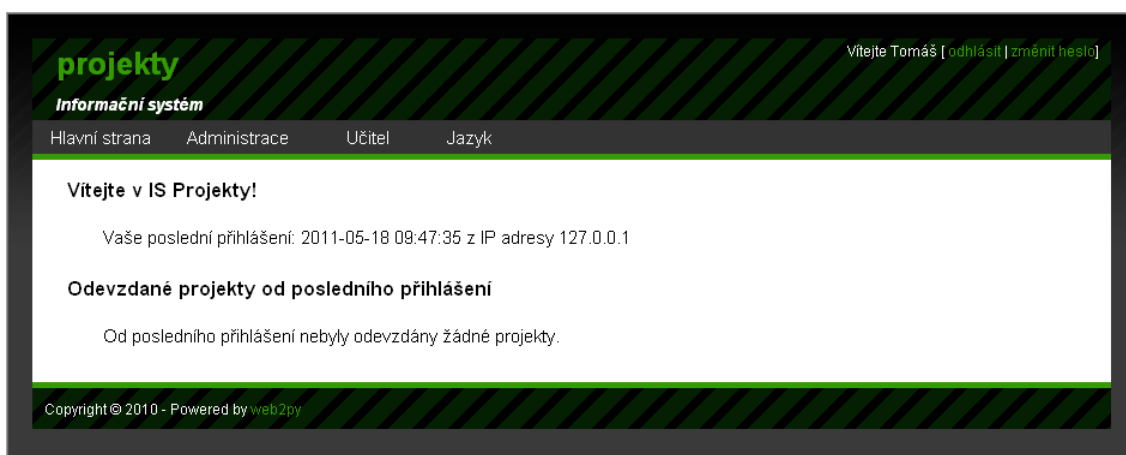
Seznam vyučování zpřístupňuje po filtraci podle předmětu všechna vyučování daného předmětu. V zobrazení vyučování lze měnit vyučujícího, zobrazit seznam studentů daného vyučování a přiřadit či odebírat zapsané studenty z vyučování. Při zapisování studentů do vyučování je možné využít filtr a omezit výsledky vyhledávání podle příjmení studenta.

B.2 Modul Učitel

Struktura modulu Učitel:

- Projekty
 - Seznam projektů
 - Nový projekt
- Docházka
- Souhrn hodnocení

Po přihlášení uživatele s rolí **učitel** se načte hlavní strana (viz. obr. B.6) s časem a IP adresou posledního přihlášení uživatele a také projekty, které byly odevzdány od posledního přihlášení učitele. V seznamu figurují i projekty, které nebyly odevzdány v termínu (tyto projekty jsou barevně odlišeny od ostatních zadaných projektů).



Obr. B.6: Hlavní strana po přihlášení učitele


B.2.1 Projekty

Projektům jsou v systému přiřazeny následující stavy:

- Vytvořený
- Zadaný
- Odevzdáný
- Ohodnocený

Použití jednotlivých stavů bude vysvětleno v následujícím textu.

Kliknutím na odkaz **Seznam projektů** dojde k načtení filtru pro omezení výsledků vyhledávání, v tomto případě lze filtrovat podle názvu projektu, jeho stavu nebo také podle předmětu. Po potvrzení formuláře je nahrán seznam projektů (viz. obr. B.7) podle zadaných parametrů filtru. Stavy projektů jsou pro přehlednost v seznamu barevně odlišeny. K editaci projektu se uživatel dostane kliknutím na ID projektu.



The screenshot shows a web application titled 'projekty' with a subtitle 'Informační systém'. The user is logged in as 'Vítejte Tomáš' with links to 'odhlásit' and 'změnit heslo'. The navigation bar includes 'Hlavní strana', 'Administrace', 'Učitel', and 'Jazyk'. The main content area is titled 'Seznam projektů' and displays a table of projects. The table has five columns: ID, Název projektu, Termin odevzdání, Stav, and Hodnocení. The projects are listed with their respective IDs, names, deadlines, statuses, and scores. The status 'Ohodnocený' is highlighted in green, 'Vytvořený' in grey, 'Odevzdáný' in blue, and 'Zadaný' in red. Below the table, it shows 'Strana 1 | Položky 1 - 10 (10 celkem)' and a dropdown menu for 'Záznamů na stranu: 20' with a 'Nastavit' button. The footer contains 'Copyright © 2010 - Powered by web2py'.

ID	Název projektu	Termin odevzdání	Stav	Hodnocení
14	Šifrování textu	2011-02-28 17:47:00	Ohodnocený	20
15	Kryptografické protokoly	2011-07-07 15:18:46	Ohodnocený	2
16	Zabezpečení OS	2011-03-02 17:50:22	Ohodnocený	51
17	Instalace CentOS	2011-05-02 19:37:58	Vytvořený	0
21	Implementace QoS	2011-03-31 14:34:37	Ohodnocený	10
22	Anonymní autentizace	2011-04-13 14:36:48	Odevzdáný	0
23	Implementace IPv6	2011-04-13 13:37:28	Zadaný	0
24	Test emailu	2011-05-18 17:52:50	Ohodnocený	10
25	Návrh topologie sítě	2011-05-11 15:36:30	Zadaný	0
28	Automatická instalace Debian	2011-05-20 18:44:59	Zadaný	0

Obr. B.7: Seznam projektů

Vytvoření nového projektu lze provést pomocí odkazu **Nový projekt**. Zadáním požadovaných údajů (předmět, název projektu, zadání a termín odevzdání) a odesláním formuláře je projekt vytvořen a současně se projektu přiřadí stav **Vytvořený**, následně je uživatel přesměrován na editaci právě vytvořeného projektu (viz. obr. B.8). Stav projektu **Vytvořený** je jediným stavem, kdy se studentovi projekt nezobrazí ve výpisu projektů. Slouží hlavně k přípravě projektu, jeho zadání nebo nastavení

virtuálních strojů. V tomto stavu lze projekt také odstranit, v ostatních stavech odstranění projektu již možné není. V pravé dolní části lze po zadání řešitelů změnit

projekty
Informační systém

Vítejte Tomáš [odhlásit] [změnit heslo]

Hlavní strana Administrace Učitel Jazyk

Editace projektu

Id: 28

Název projektu: Automatická instalace Debian **Předmět MNSB**

Zadání: Nainstalujte operační systém Debian.
<http://www.debian.org/releases/stable/i386/index.html.cs>
 Dokument obsahuje návod na instalaci systému Debian GNU/Linux 6.0 (kódové označení „squeeze“), pro počítače 32-bit PC („i386“). Kromě návodu zde naleznete odkazy

Termín odevzdání: 2011-05-20 18:44:59

Odeslat

2 kometářů | 5 virtuálních strojů | 1 souborů

Řešitelé projektu

Stav projektu: Zadaný

ID	Jméno	Příjmení	E-mail	Souhlasí s odevzdáním
43	Jiří	Havel	xxhavel00@stud.feec.vutbr.cz	Ne
41	Martin	Procházka	user@user.cz	Ne

Přidat řešitele

Copyright © 2010 - Powered by web2py

Obr. B.8: Editace projektu

stav projektu na **Zadaný**. Před samotnou změnou stavu ještě probíhají dvě kontroly:

- přiřazení minimálně jednoho řešitele,
- termín odevzdání nesmí být zvolen z minulosti.

Pokud jsou obě podmínky splněny, dojde ke změně stavu a řešitelům projektu je odeslána e-mailová zpráva o zadání nového projektu k řešení.

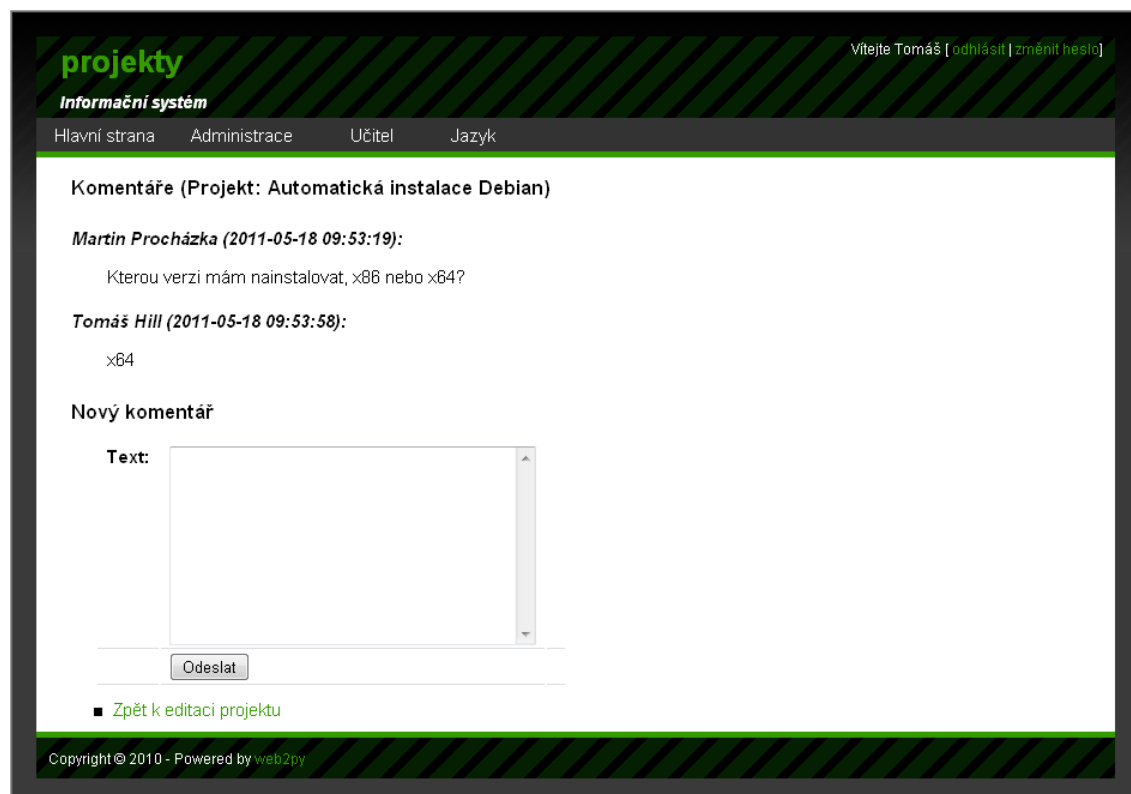
Ve stavu **Zadaný** již nelze projekt odstranit, nicméně stále lze změnit název projektu, jeho zadání a termín odevzdání stejně tak jako přidávat či odebírat řešitele. Tyto možnosti se automaticky zablokuji po odevzdání projektu řešitelem (resp. řešiteli) a poté lze projekt pouze ohodnotit.

Hodnocení projektu se provádí kliknutím na odkaz **Hodnotit projekt**, projekt lze hodnotit pouze pokud jej řešitel (příp. řešitelé) odevzdá nebo pokud není dodržen termín odevzdání. K hodnocení je také možné připojit slovní komentář, který se spolu s body odesílá e-mailem v okamžiku potvrzení hodnocení všem řešitelům

projektu. Spolu s odesláním hodnocení se stav projektu automaticky změní na konečný stav **Ohodnocený**.

U projektu se nacházejí ještě další tři odkazy, komentáře, virtuální stroje a soubory (odkazy jsou vždy doplněny o počet, tzn. např **5 virtuálních strojů**).

Komentáře (viz. obr. B.9) slouží ke komunikaci mezi vyučujícím a řešiteli projektu. S každým odesláním nového komentáře se odesílá e-mailová zpráva všem řešitelům a také zadavateli projektu.



Obr. B.9: Komentáře k projektu

Položka **virtuální stroje** zobrazuje seznam virtuálních strojů přiřazených k projektu a jejich stav (viz. obr. B.10). Učitel má možnost virtuální stroje spouštět, zastavovat nebo úplně odstranit, v případě manuální instalace stroje také dokončit instalaci (změna stavu v databázi a odpojení obrazu CD). Vytvořit virtuální stroj lze pomocí odkazu **Nový virtuální stroj** (viz. obr. B.11), kde učitel vyplní název nového virtuálního stroje, zvolí šablonu domény, šablonu pevného disku, zvolí obraz disku CD který má být připojen k instalaci a také šablonu instalace (manuální nebo automatická).

projekty

Informační systém

Vítejte Tomáš [[odhlásit](#)] [[změnit heslo](#)]

[Hlavní strana](#)
[Administrace](#)
[Učitel](#)
[Jazyk](#)

Virtuální stroje (projekt: Automatická instalace Debian)

Jméno	Stav	Start/Stop	VNC adresa	VNC port	Smazat
instalace_debian	MANUAL_INSTALL (dokončit)	Stop	147.229.151.221	5901	Smazat
debian_vzor2	RUNNING	Stop	147.229.151.221	5900	Smazat
debian_vzor3	INSTALLATION_RUNNING	Stop	147.229.151.221	5902	Smazat
debian_vzor	STOPPED	Start	nedostupné	nedostupné	Smazat
debian_vzor4	READY_TO_INSTALL	nedostupné	nedostupné	nedostupné	nedostupné

■ Nový virtuální stroj

■ Zpět k editaci projektu

Copyright © 2010 - Powered by [web2py](#)

Obr. B.10: Virtuální stroje

projekty

Informační systém

Vítejte Tomáš [[odhlásit](#)] [[změnit heslo](#)]

[Hlavní strana](#)
[Administrace](#)
[Učitel](#)
[Jazyk](#)

Nový virtuální stroj (projekt: Automatická instalace Debian)

Jméno:

Šablona domény:

basic_domain

Šablona disku:

volume_1.5GB

Obraz disku CD:

debian-506-i386-netinst.iso

Šablona instalace:

basic_manual_install_centos

Vytvořit

Copyright © 2010 - Powered by [web2py](#)

Obr. B.11: Vytvoření nového virtuálního stroje

B.2.2 Docházka

Sekce **Docházka** umožňuje evidenci docházky studentů do vyučování a také umožňuje udělovat studentům bodové hodnocení za splnění úkolů v každé hodině.

Po kliknutí na odkaz **Docházka** dojde k načtení filtru, kde učitel zvolí předmět a týden výuky, ve kterém chce evidovat docházku. Odesláním formuláře se vypíše seznam všech hodin, které přihlášený učitel v daném předmětu vyučuje. Zvolením vyučování pomocí jeho ID se načte seznam studentů zapsaných do daného vyučování, změna docházky se provádí kliknutím na ID studenta, kde lze zaznamenat zda

student byl na vyučování přítomen a také jeho bodové hodnocení (viz. obr. B.12). Potvrzením formuláře dojde k načtení dalšího studenta v seznamu a učitel opět může vyplnit docházku a přidělit studentovi body. Postup se opakuje dokud učitel nedosáhne konce seznamu.



Obr. B.12: Evidence docházky

B.2.3 Souhrn hodnocení

Souhrn hodnocení slouží k zobrazení hodnocení studentů, hodnocení je rozděleno podle vyučování do kterých jsou studenti zapsáni. Odkazem **Souhrn hodnocení** je načten filtr pro volbu předmětu a vyučování. Pro uživatele s rolí **správce** je dostupná volba **Pouze moje vyučování**, pomocí této volby lze zobrazit souhrny hodnocení všech vyučování, tzn. i těch, ve kterých není přihlášený uživatel zapsán jako učitel. Potvrzením filtru je zobrazen seznam studentů a jejich hodnocení (viz. obr. B.13)

B.3 Modul Student

Struktura modulu Student:

- Moje projekty
- Hodnocení

Přihlášením uživatele s rolí **student** dojde k načtení hlavní strany (viz. obr. B.14), kde je zobrazen čas a IP adresa posledního přihlášení uživatele a také se zobrazí seznam projektů na kterých student pracuje (projekty, které jsou ve stavu „Zadaný“). K zobrazení projektu se uživatel dostane kliknutím na jeho ID.

projekty

Informační systém

Vítejte Tomáš [odhlásit] [změnit heslo]

Hlavní strana
Administrace
Učitel
Jazyk

Souhrn hodnocení

Předmět MBIS (Po 14:00-15:40), podmínky pro získání zápočtu:

- 2 docházek na cvičeních
- 20 bodů z projektů

Jméno	Příjmení	Bodů celkem	Zápočet	Počet docházek	Body z projektů	Body ve cvičení
Luboš	Hanák	5	Ne	2	0	5
Silas	Botwin	30	Ne	1	20	10
Conrad	Shepard	10	Ne	1	0	10
Left	Join	0	Ne	0	0	0
Tim	Scottson	66	Ano	2	61	5
Isabelle	Hodes	20	Ne	1	20	0
Martin	Procházka	10	Ne	0	10	0

Strana 1 | Položky 1 - 7 (7 celkem)

Záznamů na stranu:

Copyright © 2010 - Powered by web2py

Obr. B.13: Souhrn hodnocení

projekty

Informační systém

Vítejte Martin [odhlásit] [změnit heslo]

Hlavní strana
Moje projekty
Hodnocení
Jazyk

Vítejte v IS Projekty!

Vaše poslední přihlášení: 2011-05-18 09:43:41 z IP adresy 127.0.0.1

Vyučování ve kterých jste registrován

Předmět	Vyučování	Číslo místnosti	Učitel
MNSB	St 14:00-15:40	PA-128	Hodes Celia
MBIS	Po 14:00-15:40	PA-127	Hill Tomáš

Projekty na kterých pracujete

ID	Název projektu	Termín odevzdání
28	Automatická instalace Debian	2011-05-20 18:44:59

Copyright © 2010 - Powered by web2py

Obr. B.14: Hlavní strana po přihlášení studenta

B.3.1 Moje projekty

Odkaz **Moje projekty** načte filtr, kterým lze filtrovat projekty podle předmětu (není však povinné). Po odeslání filtru se zobrazí seznam všech uživatelských projektů. Ke konkrétnímu projektu se uživatel dostane opět kliknutím na ID projektu.

V zobrazení projektu se nachází kromě názvu projektu, zadání a termínu odevzdání také odkazy na diskuzi (komentáře) k projektu, přiřazené virtuální stroje a odkaz na soubory, které jsou k projektu na serveru nahrány (viz. obr. B.15). V další části zobrazení projektu jsou zobrazeni řešitelé projektu (pokud student neřeší projekt samostatně), stav projektu a také odkaz pro odevzdání projektu.

The screenshot shows a web application titled 'projekty' with a green and black striped header. The user is logged in as 'Vítejte Martin' with links to 'odhlásit' and 'změnit heslo'. The navigation bar includes 'Hlavní strana', 'Moje projekty', 'Hodnocení', and 'Jazyk'. The main content area is titled 'Zobrazení projektu' and displays details for a project named 'Automatická instalace Debian'. The project description mentions installing the Debian GNU/Linux 6.0 'squeeze' system on a 32-bit PC. The submission deadline is '2011-05-20 18:44:59' and the creator is 'Tomáš Hill'. Below this, it shows '2 komentářů | 3 virtuálních strojů | 1 souborů'. A table lists the project solvers ('Řešitelé projektu') with columns for name, surname, email, and agreement status. The status is currently 'Zadaný' (Assigned), and there is a link to 'Souhlasím s odevzdáním' (I agree to submit). The footer indicates 'Copyright © 2010 - Powered by web2py'.

projekty
Vítejte Martin [odhlásit] [změnit heslo]

Informační systém

Hlavní strana Moje projekty Hodnocení Jazyk

Zobrazení projektu

Název projektu: Automatická instalace Debian

Zadání: Nainstalujte operační systém Debian. <http://www.debian.org/releases/stable/i386/index.html.cs>
Dokument obsahuje návod na instalaci systému Debian GNU/Linux 6.0 (kódové označení „squeeze“), pro počítače 32-bit PC („i386“). Kromě návodu zde naleznete odkazy na další dokumentaci, která vám pomůže s detailnějším nastavením a vyladěním nového systému.

Termín odevzdání: 2011-05-20 18:44:59

Zadal: Tomáš Hill

2 komentářů | 3 virtuálních strojů | 1 souborů

Řešitelé projektu

Jméno	Příjmení	E-mail	Souhlasí s odevzdáním
Jiří	Havel	xxhavel00@stud.feec.vutbr.cz	Ne
Martin	Procházka	user@user.cz	Ne

Stav projektu: Zadaný

Změnit stav: [Souhlasím s odevzdáním](#)

Copyright © 2010 - Powered by web2py

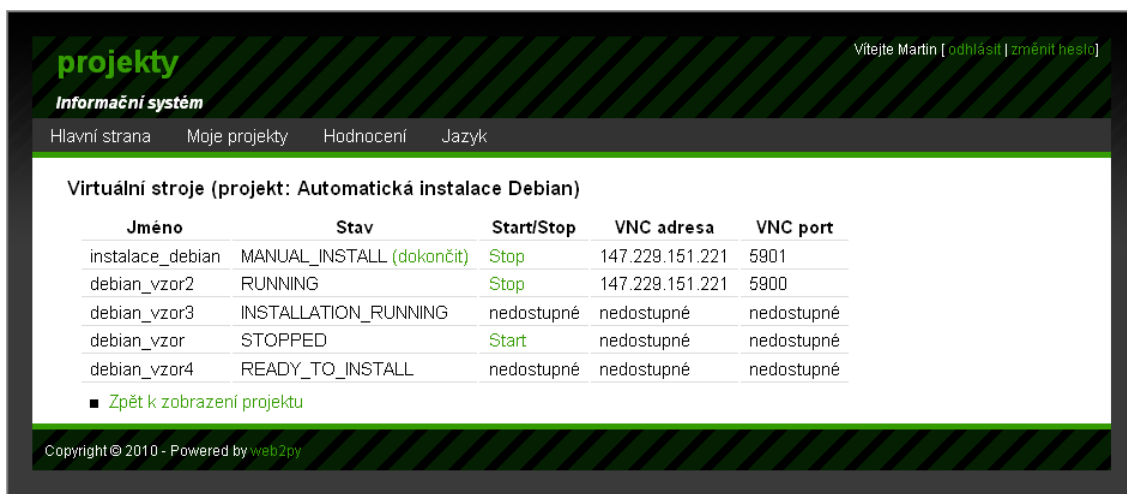
Obr. B.15: Seznam projektů studenta

Pokud projekt řeší více než jeden uživatel, je odevzdání řešeno „souhlasem s odevzdáním“. Svůj souhlas s odevzdáním může uživatel odvolat, ale pouze pokud minimálně jeden uživatel souhlas ještě nevyjádřil. Jestliže s odevzdáním souhlasí všichni uživatelé, pak je projekt automaticky odevzdán a tento stav již nelze změnit. Je-li řešitelem pouze jeden uživatel, projekt je odevzdán okamžitě po kliknutí na odkaz **Odevzdat projekt**.

Vzhled a používání formulářů **komentáře**, **virtuální stroje** a **soubory** je shodné s formuláři v modulu **Učitel**. V zobrazení virtuálních strojů jsou studentovi na rozdíl od učitele skryty adresy virtuálních strojů, u kterých probíhá automatická instalace a také možnost virtuální stroj odstranit. Student má možnost virtuální stroje spouštět nebo zastavovat, případně dokončit manuální instalaci (změna stavu stroje v databázi a odpojení virtuálního obrazu CD, viz. obr. B.16).

Odevzdání projektu, případně vyjádření souhlasu s odevzdáním v případě více řešitelů, podléhá kontrole termínu odevzdání. Jestliže se uživatel pokouší odevzdat

projekt po termínu odevzdání, není mu to dovoleno, stejně tak není dovoleno uživateli nahrávat soubory na server po termínu odevzdání projektu.



The screenshot shows a web application titled 'projekty' with a subtitle 'Informační systém'. The user is logged in as 'Vítejte Martin' with links to 'odhlásit' and 'změnit heslo'. The navigation bar includes 'Hlavní strana', 'Moje projekty', 'Hodnocení', and 'Jazyk'. The main content area is titled 'Virtuální stroje (projekt: Automatická instalace Debian)' and contains a table with the following data:

Jméno	Stav	Start/Stop	VNC adresa	VNC port
instalace_debian	MANUAL_INSTALL (dokončit)	Stop	147.229.151.221	5901
debian_vzor2	RUNNING	Stop	147.229.151.221	5900
debian_vzor3	INSTALLATION_RUNNING	nedostupné	nedostupné	nedostupné
debian_vzor	STOPPED	Start	nedostupné	nedostupné
debian_vzor4	READY_TO_INSTALL	nedostupné	nedostupné	nedostupné

Below the table is a link: ■ [Zpět k zobrazení projektu](#). The footer contains the text 'Copyright © 2010 - Powered by web2py'.

Obr. B.16: Seznam virtuálních strojů přiřazených k projektu (modul Student)

B.3.2 Hodnocení

Část **Hodnocení** po zvolení předmětu zobrazí studentovi souhrn hodnocení daného předmětu (viz. obr. B.17), tzn. zda student obdrží zápočet (případně důvody neudělení zápočtu), souhrn docházky do vyučování a souhrn všech projektů daného předmětu.

C INSTALACE FRAMEWORKU WEB2PY A WEBOVÉ APLIKACE

Tato příloha popisuje instalaci frameworku Web2py a následně také instalaci samotné webové aplikace. Postup instalace byl otestován na operačním systému Microsoft Windows XP Professional a linuxové distribuci Fedora 14.

Popis instalace předpokládá běh webové aplikace a databázového stroje na jednom stroji. Na disku CD přiloženém k diplomové práci se nachází všechny balíčky a aplikace v nejnovějších verzích v době odevzdávání práce, u každého balíčku uvedeném v postupu instalace je také uveden URL odkud lze stáhnout nejnovější verze balíčků.

C.1 Instalace na operačním systému

Microsoft Windows

V operačním systému Microsoft Windows je nutná instalace následujících balíčků:

- `postgresql-9.0.4-1-windows.exe` - databázový stroj PostgreSQL ¹
- `python-2.7.1.exe` - implementace jazyka Python do operačního systému Microsoft Windows ²
- `psycpg2-2.4.1.win32-py2.7-pg9.0.4-release.exe` - ovladač databáze PostgreSQL ³
- `web2py_src.zip` - zdrojové kódy webového frameworku Web2py ⁴

Balíčky jsou přiloženy na disku CD ve složce `instalace/win-x86/`.

Instalace databázového stroje PostgreSQL se provádí přes dialogová okna instalátoru. Databázový stroj je standardně nainstalován do složky

`C:\Program Files\PostgreSQL\9.0`, data jsou uchována ve složce

`C:\Program Files\PostgreSQL\9.0\data`. V průběhu instalace lze také zvolit port na kterém bude databázový stroj naslouchat, ve výchozím nastavení stroj komunikuje na portu 5432. Instalátor také vyzve uživatele k zadání administrátorského hesla a k výběru výchozí jazykové sady.

Operační systémy Microsoft Windows neobsahují implementaci jazyka Python, proto je nutná instalace balíčku pro podporu jazyka Python. Instalace probíhá opět prostřednictvím dialogových oken instalátoru. Instalátor vyžaduje pouze výběr umístění instalovaných souborů, standardně `C:\Python27`.

¹<http://www.postgresql.org/download/windows>

²<http://www.python.org/download/>

³<http://www.stickpeople.com/projects/python/win-psycpg>

⁴<http://web2py.com/examples/default/download>

Pro propojení databázového stroje PostgreSQL a frameworku Web2py, resp. obecně propojení s implementací jazyka Python, slouží instalátor ovladače databázového stroje `psycpg2`. Instalátor automaticky rozpozná umístění souborů jazyka Python i umístění jeho balíčků, není nutné nic nastavovat.

Posledním krokem je instalace webového frameworku Web2py. Od tvůrců frameworku je k dispozici i binární verze frameworku pro operační systémy Microsoft Windows, nicméně pro používání jiného než vestavěného databázového stroje SQLite je nutné do frameworku dohrát ovladač příslušného databázového stroje, což samozřejmě do zkompilevané binární verze nelze. Proto je nezbytné spouštět framework ze zdrojových kódů. ZIP archiv `web2py_src` se pouze rozbálí na libovolné místo na pevném disku, např. `C:\web2py`.

Po dokončení instalace lze framework spustit příkazem:

```
python C:\web2py\web2py.py -i 127.0.0.1 -p 8000 -a heslo
```

Současně dojde ke startu vestavěného webového serveru Rocket WSGI. Parametry příkazu:

- `-i`: volba IP adresy na které bude webový server naslouchat
- `-p`: volba portu na kterém bude webový server naslouchat
- `-a`: volba hesla pro přístup k webovému administračnímu rozhraní frameworku

C.2 Instalace na operačním systému Linux

V operačním systému založeném na jádře Linux je nutná instalace následujících balíčků:

- `postgresql-9.0.4-1-linux.bin` - databázový stroj PostgreSQL ⁵
- `psycpg2-2.4.1.tar.gz` - ovladač databáze PostgreSQL ⁶
- `web2py_src.zip` - zdrojové kódy webového frameworku Web2py ⁷

Balíčky jsou přiloženy na disku CD ve složce `instalace/linux-x86/`.

Na rozdíl od operačního systému Microsoft Windows je v operačních systémech s jádrem Linux jazyk Python plně implementován, není nutné jej instalovat.

Postup instalace databázového stroje se liší podle použité linuxové distribuce, pro jednoduchost byl zvolen grafický instalátor `postgresql-9.0.4-1-linux`, kde instalace probíhá stejně jako v operačním systému Microsoft Windows za pomoci dialogových oken.

```
su
chmod +x postgresql-9.0.4-1-linux.bin
./postgresql-9.0.4-1-linux
```

⁵<http://www.enterprisedb.com/products-services-training/pgdownload>

⁶<http://www.initd.org/psycpg/download>

⁷<http://web2py.com/examples/default/download>

Jediný rozdíl je v umístění souborů, v linuxu standardně `/opt/PostgreSQL/9.0` a data `opt/PostgreSQL/9.0/data`. Grafický instalátor byl otestován na několika distribucích, např. Ubuntu 6.06 a vyšší, Fedora 6 a vyšší nebo CentOS/RHEL 4 a vyšší [6]. Databázový stroj lze nainstalovat také pomocí balíčků, postup pro jednotlivé linuxové distribuce je uveden v literatuře [6].

Ovladač databázového stroje Psycopg lze nejjednodušší nainstalovat z repozitářů příkazem:

```
yum install python-psycopg2.i686
```

Opět však záleží na použité distribuci, uvedený příkaz je funkční v distribuci Fedora 14, na které byl postup testován.

Framework Web2py se shodně jako v instalaci pro operační systém Microsoft Windows pouze rozbalí na libovolné místo na disku a spustí se stejným příkazem:

```
python web2py.py -i 127.0.0.1 -p 8000 -a heslo
```

Význam parametrů zůstává stejný:

- `-i`: volba IP adresy na které bude webový server naslouchat
- `-p`: volba portu na kterém bude webový server naslouchat
- `-a`: volba hesla pro přístup k webovému administračnímu rozhraní frameworku

C.3 Instalace webové aplikace

Před nainstalováním webové aplikace je nezbytné vytvořit nového uživatele databáze (jeho prostřednictvím se framework Web2py přihlašuje k databázi) a vytvořit samotnou databázi pro uložení dat.

V operačním systému Microsoft Windows lze využít software **pgAdmin III**, který byl nainstalován zároveň s databázovým strojem **PostgreSQL**. Po připojení k databázovému stroji a zadáním hesla superuživatele, které bylo zvoleno při instalaci, lze vytvořit novou databázi v objektu **Databáze** a vytvořit nového uživatele databáze v objektu **Přihlašovací role**. Název databáze, přihlašovací jméno a heslo musí souhlasit s parametry v kódu aplikace⁸, viz. kap. 2.3.

Nastavení přihlašovacího jména a hesla spolu s názvem databáze lze nastavit v linuxové distribuci následujícími příkazy:

```
createuser -U postgres jmeno_noveho_uzivatele  
createdb -U postgres nazev_databaze
```

Webový server Rocket WSGI frameworku Web2py nyní naslouchá na adrese `127.0.0.1:8000`. Zadáním adresy `http://127.0.0.1:8000` dojde k načtení výchozí aplikace **Welcome** (viz. obr. C.1). Přihlášením do administračního rozhraní

⁸Aplikace na přiloženém disku CD je nastavena na název databáze „projekty“, přihlašovací jméno „web2py“ a heslo „123“

Hello World

Readme

- You are successfully running web2py
 - This is a copy of the scaffolding application
 - You can modify this application and adapt it to your needs
 - [click here for the administrative interface](#)
 - [click here for online examples](#)
 - [web2py.com](#)
 - [documentation](#)
1. You visited the url </welcome/default/index>
 2. Which called the function `index()` located in the file [web2py/applications/welcome/controllers/default.py](#)
 3. The output of the file is a dictionary that was rendered by the view [web2py/applications/welcome/views/default/index.html](#)

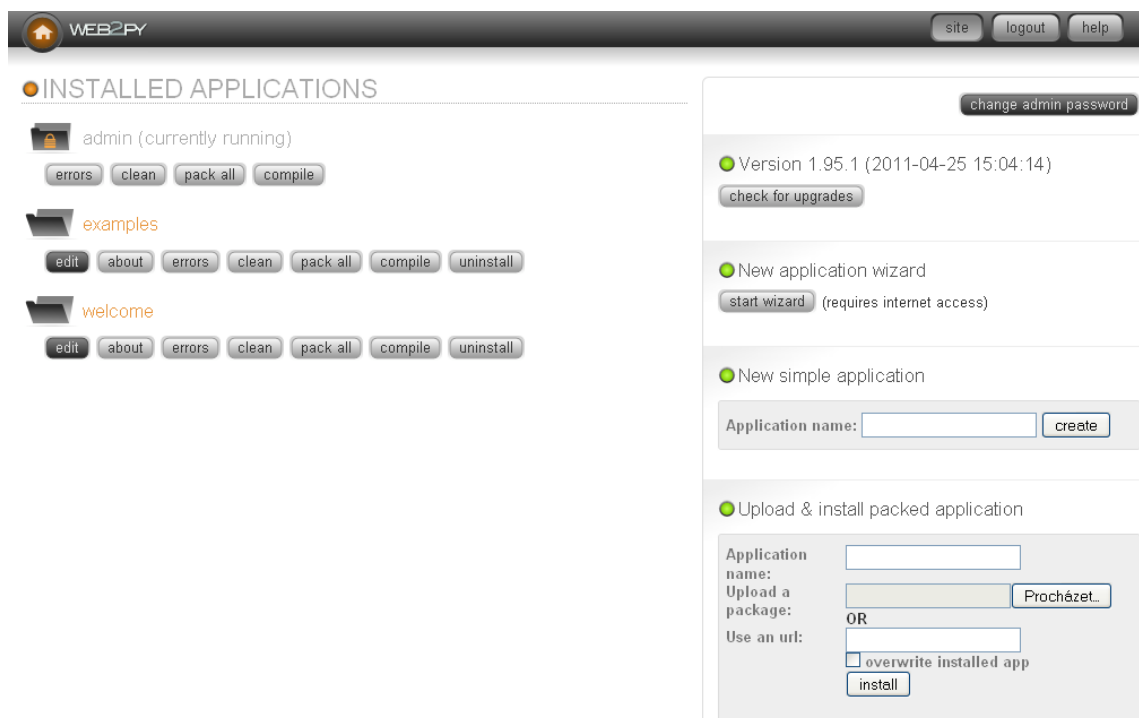


Copyright © 2010

Obr. C.1: Výchozí aplikace Welcome

(viz. obr. C.2) pomocí URL `http://127.0.0.1:8000/admin/` dojde k zobrazení nainstalovaných aplikací, v části `Upload & install packed application` je možné vybrat zabalenou aplikaci k nainstalování⁹. Po nainstalování (např. aplikace s názvem „projekty“) je možné aplikaci editovat, zobrazit chyby, odstranit sezení a chyby, zabalit k exportu, zkompileovat, případně aplikaci odstranit. Aplikace je nyní dostupná na adrese serveru a pod názvem, který byl zadán při instalaci aplikace (tzn. např. `http://127.0.0.1:8000/projekty/`). Do aplikace na přiloženém disku CD se lze přihlásit jako správce pomocí přihlašovacího jména „admin@admin.com“ a hesla „administrator951“.

⁹Zabalená aplikace na disku CD: `instalace/web2py.app.projekty.w2p`



Obr. C.2: Webové administrační rozhraní